

A NEW SIMULTANEOUS SEMI-TRAY-BY-TRAY APPROACH TO MULTICOMPONENT DISTILLATION CALCULATIONS

**PART—A
FORMULATION, PROPERTY ESTIMATION WITH NONIDEAL
THERMODYNAMICS AND DESIGN**

**A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY**

**by
SAIBAL GANGULY**

**to the
DEPARTMENT OF CHEMICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
JUNE, 1985**

CHE-1905-M-GAN-NEW

9-7 86

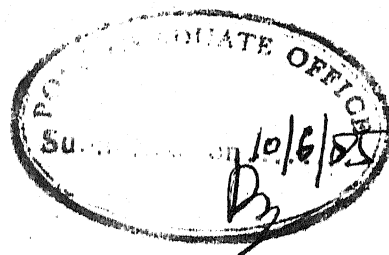
I.I.T. KANPUR
CENTRAL LIBRARY
Inv. No. A 91886

DEDICATED

TO

MY MOTHER & FATHER

My achievement is little
compared to their sacrifice



CERTIFICATE

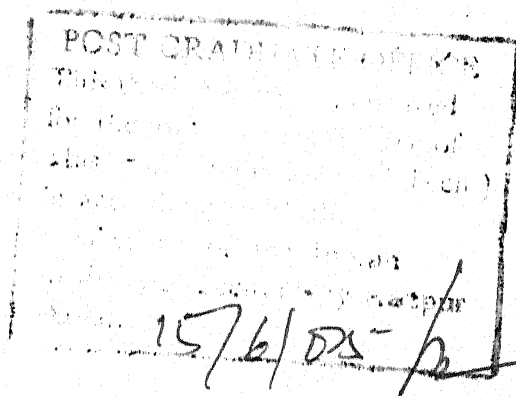
It is certified that the present work entitled "A NEW SIMULTANEOUS SEMI-TRAY-BY-TRAY APPROACH TO MULTICOMPONENT DISTILLATION CALCULATIONS - PART A" has been carried out by Mr. Saibal Ganguly under my supervision and that it has not been submitted elsewhere for a degree.

June 10, 1985

D.N. Saraf

(Dr. D.N. Saraf)
Professor

Department of Chemical Engineering
Indian Institute of Technology
Kanpur-208016, INDIA



ACKNOWLEDGEMENTS

I feel great pleasure to express my deep sense of gratitude and respect to our professor, Dr. D.N. Saraf for providing discerning guidance, valuable advice and constructive criticism throughout the course of this project work. But for his sympathetic attitude and constant inspiration, the new development could not have been accomplished. He has been more than a professor to us and I am happy that I had an opportunity to work with him.

Thanks are due to Dr. Anil Kumar, Dr. S.K. Gupta and Dr. R.P. Singh for providing valuable suggestions from time to time.

I am thankful to all my classmates particularly, to my dear friend and colleague Mr. A.K. Das for having helped me throughout this joint venture and who made my stay at IIT-Kanpur cheerful.

Lastly, but not the least, I tender my cordial thanks to Mr. B.S. Pandey for typing this thesis really at a very short notice; to Mr. A.K. Ganguly for the drawings and to Mr. D.B. Chakraborty (dada) for extending his loving help on many a difficult occasion.

Saibal Ganguly
(Author)

CONTENTS

List of Figures	vi
List of Tables	vii
Nomenclature	viii
Abstract	xii

CHAPTER

1	INTRODUCTION	1
2	LITERATURE REVIEW	4
	2.1 Advances in Design	4
	2.2 Semi-tray-by-tray Algorithm (Ohmura and Kasahara, 1978)	7
3	NEW SEMI-TRAY-BY-TRAY METHOD WITH SIMUL- TANEOUS EQUATION SOLVING (PRESENT METHOD)	20
	3.1 Singletray model	21
	3.2 Multitray model	24
	3.3 Variables selection and matrix structure	34
	3.4 Solution procedure	39
4	THERMODYNAMIC PROPERTY ESTIMATIONS FOR MULTICOMPONENT DISTILLATION CALCULATIONS	48
	4.1 Introduction	48
	4.2 Basic Property Estimation for pure components	50
	4.3 Calculation of partition coeffi- cients $K_{i,j}$	54
	4.4 Calculation of enthalpies, $h_{i,j}$ and $H_{i,j}$	62
	4.5 Calculation of Dew Point and Bubble Point	63
5	RESULTS AND DISCUSSION	76
	5.1 VLE Data Prediction	77

CHAPTER

5.2 Dew Point and Bubble Point Calculation	85
5.3 Simulation of Distillation Columns	88
5.4 Design of Distillation Columns	92
6 CONCLUSION AND RECOMMENDATIONS	95
REFERENCES	97

APPENDIX

A Flowchart for Lewis Matheson Algorithm (1932)	100
B Flowchart for Ricker and Grens Algorithms (1974)	102
C Flowchart for Ohmura and Kasahara Algorithm (1978)	103
D Flowchart for Present Method	104
E Method to Obtain Solution of Cubic Equation (Perry, 1963)	105
F Program Listing	106
G Detailed Output Listing	

LIST OF FIGURES

FIGURE

1	Schematic representation of a complex multistage distillation column	9
2	Schematic representation of a general singletray unit	11
3	Schematic representation of a general multitray unit	11
4	Structural pattern in the submatrices A_i , B_i and C_i of the overall Jacobian matrix: singletray unit	42
5	Structural pattern in the submatrices A_i , B_i and C_i of the overall Jacobian matrix multitray unit	45

LIST OF TABLES

TABLE

1	Possible system equations for Distillation columns	17
2	Estimated basic physical properties for halosubstituted silane compounds	65
3	The Soave-Redlich-Kwong Equation of State	66
4	The Peng-Robinson Equation of State	69
5	The UNIFAC Group contribution method	72
6	Flowchart for Dew Point Estimation	74
7	Flowchart for Bubble Point Calculation	75

NOMENCLATURE

$A_{i,j}$: absorption factor for jth component on ith tray
\bar{A}_i	: submatrices in the Jacobian Matrix
b_j	: constant in the equations of state
b_j	: constant in the equation of state for jth component
$b_{i,j}$: sidedraw for jth component on ith tray in stripping section
\bar{B}_i	: submatrices in Jacobian Matrix
C	: total number of components
\bar{C}_i	submatrices in Jacobian Matrix
$d_{i,j}$: sidedraw for jth component on ith tray in enriching section
E_i	: enthalpy discrepancy function for ith unit
$E_{i,j}$: Murphree plate efficiencies for jth component on ith tray or unit
f_j	: fugacity for jth component
$F_{i,j}$: molar feed flow rate for jth component on ith unit
\bar{F}_i	: column matrix of discrepancy functions
$h_{i,j}$: liquid enthalpy for jth component on ith unit
$H_{i,j}$: vapour enthalpy for jth component on ith unit
i	: stage or unit number
I	: the iteration number
j	: component number
\bar{J}	: Jacobian Matrix
$K_{i,j}$: partition coefficient or equilibrium constant for jth component on ith tray
$l_{i,j}$: liquid component flowrate for jth component on ith unit

- $L_{i,m,j}$: liquid component flowrate for jth component on the mth tray inside the ith multitray (B-type) unit
 L_i : total liquid flow rate on ith unit
 $L'_{i,m}$: total liquid flow rate on mth tray inside the ith multitray (B-type) unit
 $M_{i,j}$: component material balance discrepancies for jth component on ith unit
 M_i : number of stages inside the ith multitray (B-type) unit
 M_j : molecular weight for the jth component
 N : total number of trays or units in a column
 P : pressure
 $P_{c,j}$: critical pressure for jth component
 P_r : reduced pressure ($=P/P_c$)
 $P_{VP,j}$: vapour pressure for jth component
 $Q_{i,j}$: equilibrium discrepancy for jth component on ith unit
 Q_i^* : heat given in or taken out from the ith stage
 R : checking parameter in cubic equation solution, also, universal gas constant
 s_i : fraction of material (passing to the next successive stage) which is taken as a side-draw from ith unit
 $S_{i,j}$: stripping factor for jth component on ith tray ($=1/A_{i,j}$)
 $S'_{i,m,j}$: stripping factor for jth component on mth tray inside ith multitray (B-type) unit
 $S_{e,i,j}$: effective Edmister stripping factor for jth component on ith multitray (B-type) unit
 $S''_{i,j}$: effective second factor for Edmister double-factor equation for jth component on ith multitray (B-type) unit.

- T_B, T_D : bubble point and dew point temperatures
 T_i : temperature on ith stage
 $T_{i,m}$: Temperature on mth stage inside ith multitray (B-type) unit
 $T_{c,j}$: critical temperature for jth component
 $T_{b,j}$: boiling point temperature for jth component
 $v_{i,j}$: component vapour flow rate for jth component on ith unit
 $v_{i,m,j}$: component vapour flow rate for jth component on the mth tray inside the ith multitray (B-type) unit.
 V_i : total vapour flow rate for the ith unit
 $V_{i,m}$: total vapour flowrate for the mth tray inside the ith multitray (B-type) unit
 V : volume
 $V_{c,j}$: critical volume for jth component
 x_j : liquid phase mole fractions for jth component
 \bar{X} : column matrix for independent variables
 $\bar{\bar{X}}$: column matrix for the correction vectors
 y_j : vapour phase mole fractions for jth component
 z_j : mole fraction parameter (either liquid or vapour)
 $Z_{c,j}$: critical compressibility for jth component
 Z : compressibility factor

Greek Letters

- c : Riedel factor
 ϕ : optimal fraction by Golden Section Search

- γ_j : activity coefficient for jth component
- ϵ : tolerance limit
- ϕ_j : fugacity coefficient for jth component
- μ_j : chemical potential for jth component
- z_j : acentric factor

Subscripts

- i : tray or unit number
- j : component number
- k : running index for variables
- m : tray number inside multitray (B-type) unit

Superscripts

- F : refers to feed
- I : refers to iteration number
- L : refers to liquid phase
- V : refers to vapour phase

ABSTRACT

The semi-tray-by-tray method for multistage, multi-component distillation column calculations provides the advantage of solving a variety of design and simulation problems with different end specifications to the desired degree of accuracy. The present work aims at overcoming the shortcomings of the available semi-tray-by-tray method by combining the tray-by-tray model of Naphtali-Sandholm and the shortcut model of Edmister with a simultaneous equation solving procedure. To promote convergence, the single variable Golden Section Optimization technique has been coupled with the Newton-Raphson method of solving simultaneous nonlinear algebraic equations. The linear flow profile approximation in the earlier semi-tray-by-tray method has been removed and the requisite flowrates are obtained by solving enthalpy balance equations along with equilibrium relations. Also, a more rigorous double stripping factor Edmister equation has been used as derived from the system of equations of the multitray model. The number of dependent variables has been reduced to a minimum still retaining the block tridiagonal structure in the matrix which permits the use of efficient block Thomas algorithm. The generalized algorithm can be used both for design as well as rating of simple or complex columns. The program provides a choice of condenser types, tower configuration

and plate efficiencies. Ideal as well as nonideal thermodynamic property evaluation procedures have been incorporated in the program. In case of nonideal mixtures, the user has a choice between SRK and Peng-Robinson methods or he can provide any other method of his choice. It also incorporates flexibility with respect to a wide range of end specifications. The CPU time and memory requirements are almost independent of the number of trays.

CHAPTER 1

INTRODUCTION

Multicomponent distillation plays a very important role in the separation process technology. Two different types of problems arise - the simulation or rating problem and the design problem. In the rating problems, the separation that can be attained with a given column configuration under given conditions, is to be found whereas in the design problems, column configuration and operation to meet the desired separation specifications are to be determined.

An efficient generalised method for solving multicomponent multistage distillation problems should essentially have the following features - (i) Simplicity of formulation, (ii) flexibility of application - i.e., capability to handle problems with various complex tower configurations, feed conditions and locations, variety of end specifications and various types of ~~condensers~~ and reboilers taking into account nonideality with respect to thermodynamics and stage efficiencies, (iii) Sound numerical technique ensuring stable and fast convergence having less memory and CPU time requirement, (iv) good structure and conditioning of the Jacobian matrix and (v) capability to handle both the design and simulation problems with desired degree of accuracy.

Variety of approaches have been suggested for handling multicomponent, multistage distillation calculations such as short cut, tray-by-tray and semi-tray-by-tray methods. In the short cut method (Underwood 1932; Fenske 1932; Edmister 1943 etc.), empirical correlations or a set of assumptions are used to solve the problem with comparatively less efforts. In the tray-by-tray methods either a sequential calculation is undertaken or the whole set of exact system equations is solved simultaneously by an efficient numerical algorithm. This is a rigorous method of solution and consumes larger CPU time and memory. Some of the important methods in this class have been discussed in a companion report (Das, 1985). The semi-tray-by-tray approach is a novel concept given by Ohmura and Kasahara (1978). It maximizes the merits of both the above mentioned types of methods and still reasonably retains the accuracy of the tray-by-tray methods.

The semi-tray-by-tray method, originally given by Ohmura and Kasahara, had certain shortcomings and to overcome these, in the present study the semi-tray-by-tray procedure has been modified to include the tray-by-tray model of Naphtali and Sandholm and the multitray model of Edmister and a simultaneous equation solving approach has been used. The detailed formulation of the present method has been discussed in Chapter 3. Its applications to various end specifications

have been discussed in a companion report (Das, 1985). In Chapter 4 the incorporation of non-ideal thermodynamics has been discussed. Some nearly-ideal and highly-nonideal systems have been studied and the results have been cited in Chapter 5. A brief description of the nonideal subroutines in the computer package has also been provided.

CHAPTER 2

LITERATURE REVIEW

A variety of methods have been reported in the literature to obtain the rigorous solutions for the two classes of problems namely - design and simulation. In this review some of the prominent methods for design will be outlined. The literature on simulation methods have been reviewed elsewhere (Das, 1985).

2.1 Advances in Design:

Before attempting to review the algorithms available in literature to handle the design problem, it should be made explicit that most of the so called "design methods" do not really conform to design and can only be economically and favourably used when the number of equilibrium stages in the tower and the feed locations are known. Exact solution of the "design problem" was first systematically approached by Lewis and Matheson (1932). They described a stage to stage calculation procedure to iteratively find the number of trays in the enriching and stripping sections which would allow a desired distillate product. This method assumes a product distribution and proceeds to calculate the component mole fractions at each stage simultaneously from the top and bottom of the tower. The calculation is terminated when the

feed mole fractions are matched. An empirical equation is used to generate the new corrected product distribution for the next iteration. A block diagram for this method has been provided in Appendix A. This method, however, has a number of shortcomings, which severely restrict its general application. It is essentially restricted to single feed distillation problems and difficulties arise in problems with extremes of composition. In superfractionator type columns, the top and bottom products are extremely pure. Under such conditions the system parameters may become sensitive to minute change in product compositions. Arbitrary procedures should be set up to handle non-distributed components since the concentration are sensitive to initialization. Numerically also the sequential tray-by-tray calculations are unstable due to building up of truncation errors. The method is susceptible to divergence for larger problems with nonideal mixtures. The rate of convergence by this approach is much slower than the more recent equation solving approaches.

For a long time no new significant approach was developed in the field of distillation column design. In 1974, Ricker and Grens (1974) presented a calculation procedure which they named "Design Successive Approximation". This is essentially solving the design problem by repeated application of simulation procedures. For the efficiency of this method, the number of

column configurations examined must be small and accurate estimates of composition, flow rates and temperature must be computed at the beginning of each column calculation which makes the best use of values of previous column configurations. In an inner loop the Naphtali-Sandholm algorithm is used to generate the product distribution for an assumed number of trays and in an outer loop the tray numbers N_E and N_S are changed to meet the split specifications. The change in the total stage requirement (i.e., $N_E + N_S$) is obtained from the difference between the calculated and specified reflux rates through a linearization of the Erbar-Maddox (1961) correlation. The corrected (N_E/N_S) ratio is calculated from a secant method convergence based upon the relationship between the number of stages in either section and the change in the key component ratio over that section as given by Fenske (1932). The remaining step involves the estimation of new temperature and component flow profiles for the altered column configuration, so as to preserve as much as possible, the convergence achieved in the inner loop in previous iterations. This involves scaling the individual component flows in proportion to the changed reflux and allowing for the change in the number of stages in a section, as well as scaling the temperatures in the middle portion of the column and holding temperatures unchanged near the ends of each section. The flow chart of this algorithm has been presented in Appendix B. This method is

reasonable for simple columns with regular profiles. However, attempts to solve the design problem through repeated simulation has run into several snags when applied to design of columns with multiple-feeds or sidestreams or with unusual flow profiles. Ohmura and Kasahara (1978) proposed a new distillation calculation method utilizing the salient features of both the shortcut and tray-by-tray approach. This method can handle both design and simulation problems. Because of the novelty of the concept and since it also serves as the basis for the new proposed formulation, this method will be described separately in detail in the next section.

2.2 Semi-Tray-by-Tray Algorithm (Ohmura and Kasahara, 1978):

Naphtali and Sandholm (1971) method, while performs extremely well for operating problems, can not be used for design particularly design of complex columns with multiple feed and/or with side streams. Semi-tray-by-tray method of Ohmura and Kasahara (1978) is claimed to overcome this difficulty. This method utilizes both rigorous and short cut methods to obtain a solution close to tray-by-tray calculations. Hence the equilibrium stages are defined by two types of unit models:

A-Type Units: where conventional tray-by-tray equations are used for each single equilibrium theoretical stage.

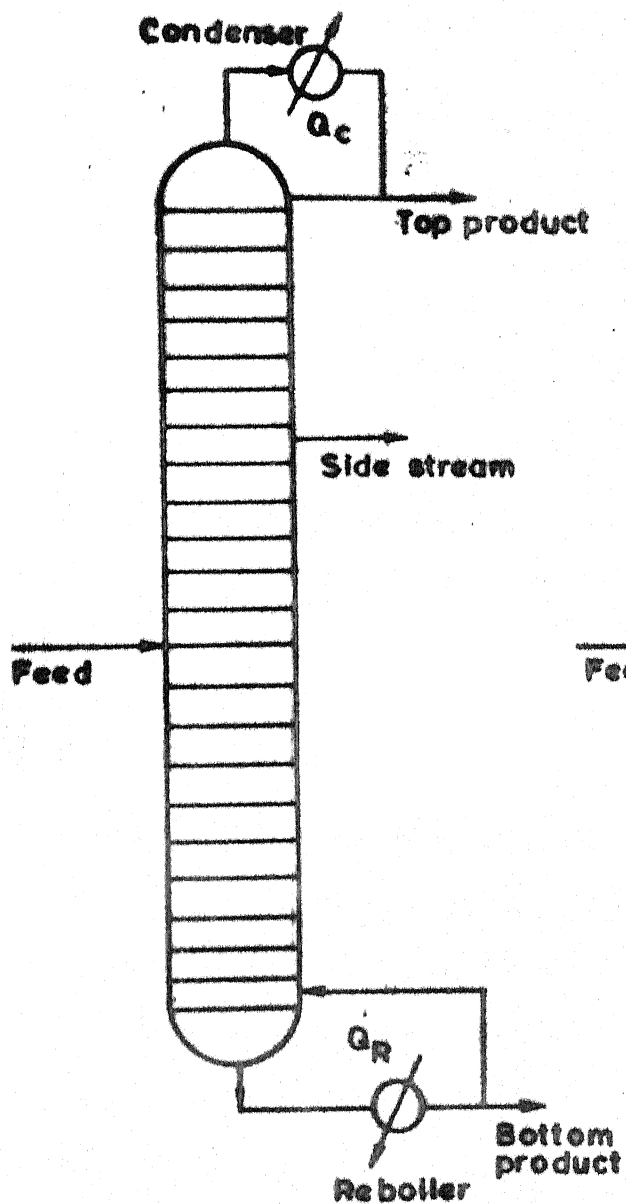
B-Type Units: where shortcut calculation method of Edmister (1943) is applied and consists of an arbitrary number of trays in each group. The following assumptions are made for these units.

1. Loads (flow rates) change linearly inside the unit.
2. Single effective stripping factor by Edmister can be used in the system equation derived.
3. Vapour and liquid leaving each tray are at bubble and dew point conditions.

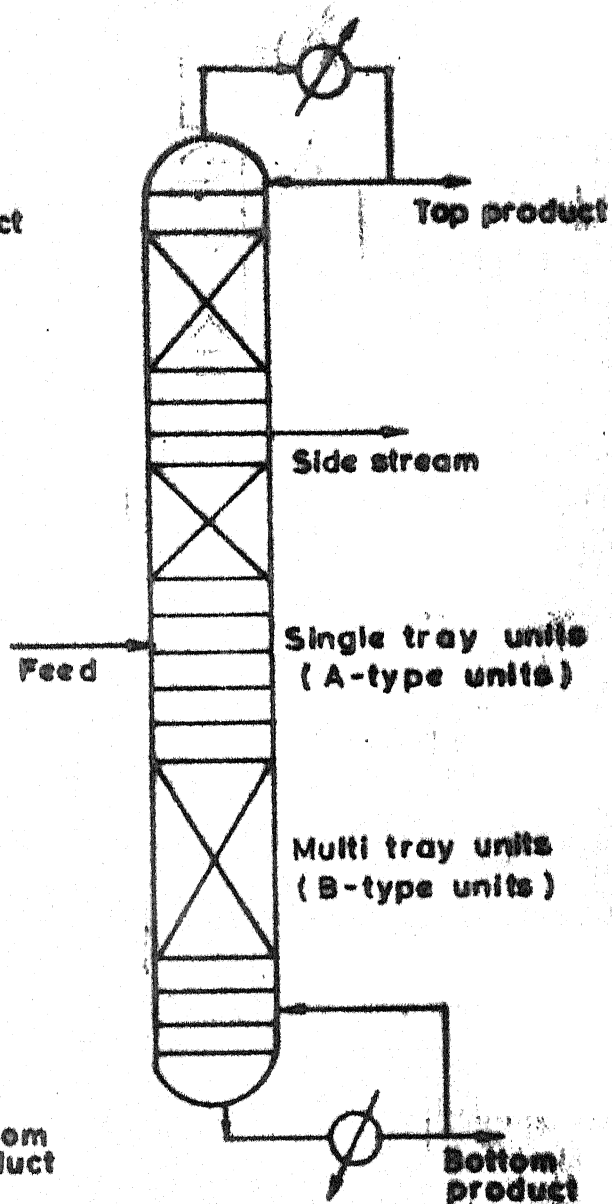
Figure 1 shows how to represent a complex distillation column as an assembly of these two types of units. For theoretical tray model or A-type unit (shown in Figure 2), the set of system equations are:

1. Combined component material balance and equilibrium relations (Equation 6 in Table 1)
2. Summation equation (Equation 4(b) in Table 1)
3. Enthalpy balance equation (Equation 5 in Table 1)

However, it should be noted that in this case i refers to the unit number and not the plate number. It is also interesting to note that in a specific column with increasing number of A-type units, the calculation results approach those of the conventional tray-by-tray method while with increasing



a) Conventional tray by tray model



(b) Semi tray by tray model
single tray and multitray

FIG. 1 SCHEMATIC REPRESENTATION OF A COMPLEX MULTISTAGE DISTILLATION COLUMN

size of the B-type units the results approach those of the shortcut method.

2.2.1 Short Cut Model or the B-type Unit:

In the B-type unit, M theoretical trays are dealt with as one model. Figure 3 shows a generalised B-type unit with the variables. The variables with dashes indicate that they belong to the inside of the B-type unit only. The system equations for the B-type unit are

(a) Overall component material balance for the B-type Unit:

$$v_{i+1,j} + l_{i-1,j} - v_{i,j} (1 + s_i^V) - l_{i,j} (1 + s_i^L) + f_{i,j}^L + f_{i,j}^V = 0 \quad (2.1)$$

(b) Overall Equilibrium Relation for the B-Type Unit:

Since in this method, an unit is being considered as a whole instead of a single tray, a relation between the outgoing vapour stream ($v_{i,j}$) and the outgoing liquid stream ($l_{i,j}$) is required. This relationship has been termed as the overall equilibrium relation for the unit. Now, for any mth tray and jth component inside the ith B-type unit, the platewise equilibrium relation is given by

$$v_{i,m,j}' = S_{i,m,j}' l_{i,m,j}' \quad (2.2)$$

$$\text{where } S_{i,m,j}' = \frac{K_{i,m,j}' v_{i,m}'}{L_{i,m}'} \quad (\text{by definition}) \quad (2.3)$$

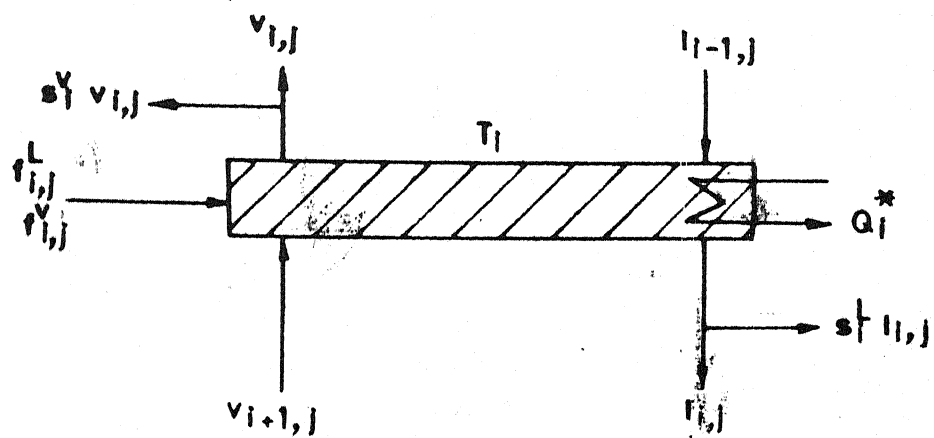


FIG. 2 SCHEMATIC REPRESENTATION OF A GENERAL SINGLETRAY UNIT

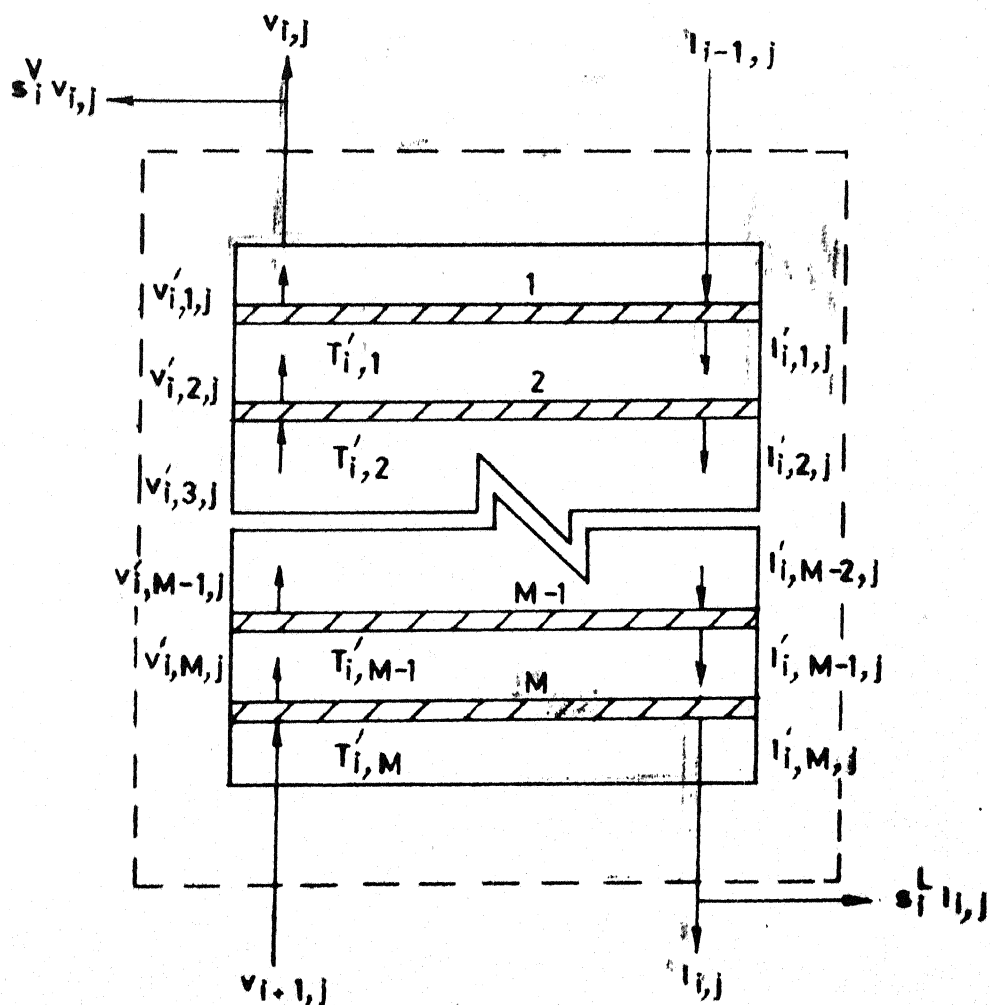


FIG. 3 SCHEMATIC REPRESENTATION OF A GENERAL MULTITRAY UNIT

The overall equilibrium relation was deduced from the component material balances and the platewise equilibrium relations by an approach similar to Edmister (1943). This deduction will be presented in Chapter 3. Edmister obtained an equation similar to equation 2.4 except an additional term which had a second stripping factor. Ohmura and Kasahara approximated Edmister's expression in terms of a single effective stripping factor without rigorously following the derivation from the relevant system equations. The final equation used by them is an approximate equation, using a single effective stripping factor of Edmister, $Se_{i,j}$, given by

$$v_{i,j} = \frac{D_{i,j} l_{i-1,j} + \pi_{i,j} (1+s_i^L) l_{i,j}}{(1+s_i^V) (1 + D_{i,j})} \quad (2.4)$$

$$\text{where } D_{i,j} = \frac{Se_{i,j} (1 - Se_{i,j}^{M_i-1})}{(1 - Se_{i,j})} \quad (2.5)$$

$$\text{and } \pi_{i,j} = (Se_{i,j})^{M_i} \quad (2.6)$$

$$Se_{i,j} = [s_{i,1,j}^i (1 + s_{i,M,j}^i) + 0.25]^{0.5} - 0.5 \quad (2.7)$$

with

$$s_{i,1,j}^i = \frac{K_{i,1,j}^i (1+s_i^V) v_i}{L_{i,1}^i} \quad (2.8)$$

and

$$s_{i,M,j}^i = \frac{K_{i,M,j}^i v_{i,M}^i}{(1+s_i^L) L_i} \quad (2.9)$$

The difficulty arises when one proceeds to calculate $L_{i,1}^I$ and $V_{i,M}^I$. Ohmura and Kasahara made a major approximation that flow profiles inside the B-type unit are linear. Thus the vapour discharge rate from the lowest tray of the model was expressed as

$$V_{i,M}^I = V_{i+1} - \frac{(1+s_i^L) \cdot L_i - L_{i,1}^I}{(M_i-1)} \quad (2.10)$$

(c) Summation equation: (Equation 4(b) in Table 1)

(d) Dew Point and Bubble Point Equations:

There are two temperatures in any B-type unit appearing in the overall equilibrium relation through the respective equilibrium constants $K_{i,1,j}^I$ and $K_{i,M,j}^I$. These top and bottom tray temperatures in the B-type unit have been defined by the respective dew point and bubble point equations as follows:

$$\frac{1}{V_i} \sum_{j=1}^C \frac{V_{i,j}}{K_{i,1,j}^I} - 1 = 0 \quad (2.11)$$

and
$$\frac{1}{L_i} \sum_{j=1}^C K_{i,M,j}^I L_{i,j} - 1 = 0 \quad (2.12)$$

(e) Enthalpy equation: (Equation 5 in Table 1)

2.2.2 Variables and Convergence Procedure:

The convergence procedure was similar to that of Tomich (1970) utilizing a two loop convergence scheme. In the first,

loop, the component flow rates are generated by using the set of combined material balances and equilibrium equations. For B-type units, a combination of equations 2.1 and 2.4 have been used. In the second loop, the enthalpy balance and the summation equations for the A-type units (Equations 5 and 4(b) in Table 1) were solved for L_i 's and T_i 's. However, for the B-type units, the summation equation, enthalpy balance, dew point and bubble point equations were solved for L_i , top and bottom temperatures ($T_{i,1}'$ and $T_{i,M}'$) and the number of trays, M_i .

The convergence technique used was simple Newton-Raphson iteration with direct inversion of the Jacobian matrix. In the first loop, however, a tridiagonal structure was present and hence Thomas algorithm could be used. The correction vector was given by

$$\Delta \vec{X} = - [\vec{J}]^{-1} \vec{F} \quad (2.13)$$

and the new variables were computed as

$$\vec{X}_{\text{NEW}} = \vec{X}_{\text{OLD}} + \Delta \vec{X} \quad (2.14)$$

Ohmura and Kasahara have cited simple examples demonstrating the application of this semi-tray-by-tray method.

2.2.3 Shortcomings of Ohmura-Kasahara Method:

Though this semi-tray-by-tray method is capable of handling simulation as well as design problems for even complex columns, it has several limitations. The method fails to handle

15

complicated problems because of its list of simplifying assumptions and numerical instability. It, thereby fails to satisfy some of the important criteria for a generalized efficient algorithm discussed in Chapter 1.

A critical analysis of the formulation shows that there is a large number of dependent variables, specially, all the componentwise vapour and liquid flowrates. The rigorous Jacobian calculation would involve finding the partial derivatives of these dependent variables with respect to the independent variables. Too many dependent variables lead to the difficulty of modulation inside the two loops specially for large problems. In a design problem neither the total number of trays nor the feed location are known. Hence the relative sizes of B-type units are also unknown. In such a situation, two consequences are often possible. A convergence may not be reached at all or if any shifting of dependent variables is possible, a distorted result may appear. A number of simplifying approximations have been made which may not always be justified. A linearity of flow profile is objectionable for towers with considerably large B-type units as also in problems with unusual profiles. Also, in this algorithm, provisions for nonideal mixtures and plate efficiencies were not made. The authors have remained silent about the flexibility of the algorithm and provision of end specifications to tackle various

difficult problems. Numerically, the algorithm seems to be poorly equipped. It lacks proper conditioning of matrix and there is no sound matrix structure to enable the use of efficient numerical algorithms. The Newton-Raphson technique, when used by itself, can lead to divergence or oscillation for complex problems or unusual cases and therefore a guiding technique to convergence should be imposed to obtain a good convergence criterion.

However, the algorithm has a very novel approach of incorporating the number of trays as independent variables. Based on this concept, a new simultaneous semi-tray-by-tray algorithm has been developed to overcome the difficulties discussed above and will be presented in the next Chapter.

TABLE 1

Possible System Equations for Distillation Columns

1. Material balance for jth component on ith tray:

$$v_{i+1,j} + l_{i-1,j} - v_{i,j}(1+s_i^V) - l_{i,j}(1+s_i^L) + f_{i,j}^L + f_{i,j}^V = 0$$

2. Total material balance for ith tray:

$$V_{i+1} + L_{i-1} - V_i(1+s_i^V) - L_i(1+s_i^L) + F_i^L + F_i^V = 0$$

3. Equilibrium relation for jth component on ith tray:

$$(a) \ y_{i,j} - K_{i,j} x_{i,j} = 0 \quad \text{or} \quad (b) \ l_{i,j} - A_{i,j} v_{i,j} = 0$$

4. Summation equation for ith tray:

$$(a) \ \sum_{j=1}^c v_{i,j} - V_i = 0 \quad \text{and} \quad (b) \ \sum_{j=1}^c l_{i,j} - L_i = 0$$

5. Enthalpy balance equation for ith tray:

$$\sum_{j=1}^c [l_{i-1,j} h_{i-1,j} + v_{i+1,j} H_{i+1,j} - (1+s_i^L) l_{i,j} h_{i,j} - (1+s_i^V) v_{i,j} H_{i,j} + f_{i,j}^L h_{i,j}^F + f_{i,j}^V H_{i,j}^F] = 0$$

6. Combined material equilibrium relation for jth component on ith tray:

$$l_{i-1,j} + s_{i+1,j} l_{i+1,j} - l_{i,j} [(1+s_i^L) + s_{i,j} (1+s_i^V)] + f_{i,j}^L + f_{i,j}^V = 0$$

7. Component material balance for i th tray including condenser reboiler:

$$(a) \frac{v_{i+1,j}}{d_{i,j}} = \left(\frac{l_{i,j}}{d_{i,j}} + 1 \right) \quad \text{and} \quad (b) \frac{v_{i+1,j}}{b_{i,j}} = \frac{l_{i,j}}{b_{i,j}} - 1$$

$$(c) \text{ Note: for condenser (i.e., } i=1): \frac{v_{2,j}}{d_{1,j}} = R + 1$$

$$(d) \text{ Note: for simple column: } d_{1,j} = \frac{f_{i,j}}{1 + \left(\frac{b_{1,j}}{d_{1,j}} \right)}$$

Notations:

$A_{i,j}$: Absorption factor for j th component on i th tray
 $(= L_{i,j}/K_{i,j} V_i)$

$b_{i,j}$: product drawn from stripping section for j th component from i th tray

$d_{i,j}$: product drawn from enriching section for j th component from i th tray.

$f_{i,j}$: feed input for j th component on i th tray in liquid condition ($f_{i,j}^L$) or in vapour state ($f_{i,j}^V$)

F_i : total feed on i th tray in liquid state (F_i^L) or in vapour state (F_i^V)

$h_{i,j}$: liquid phase enthalpy for j th component on i th tray

$H_{i,j}$: vapour phase enthalpy for j th component on i th tray

$K_{i,j}$: partition coefficient for j th component on i th tray

$l_{i,j}$: liquid flow rate for j th component from i th tray

L_i : total liquid flow rate from i th tray

s_i : side stream fraction i.e., fraction of the stream taken as side product.

$S_{i,j}$: stripping factor for j th component on i th tray ($= 1/A_{i,j}$)

$v_{i,j}$: vapour flow rate for j th component from i th tray

V_i : total vapour flow rate from i th tray

Superscripts:

F : refers to the feed

L : refers to the liquid phase

V : refers to the vapour phase.

CHAPTER 3

NEW SEMI-TRAY-BY-TRAY METHOD WITH SIMULTANEOUS EQUATION

SOLVING (PRESENT METHOD)

The semi-tray-by-tray method of Ohmura and Kasahara discussed in the previous chapter, includes a novel concept of combining the A and B type units which lead to the possibility of its use for design. The number of trays in a B-type unit can be unknown thus providing for its determination leading to design. However, there are some shortcomings discussed in the previous chapter, which restricts its general application. In the present method, Naphthali-Sandholm type efficient solution procedure has been coupled with the semi-tray-by-tray approach. Also the assumptions e.g., linear load variations inside the B-type unit etc. have been removed. We will discuss these in detail.

Figure 1 shows a multistage distillation column with single tray units (A-type units of Ohmura & Kasahara) and multitray units (B-type units of Ohmura & Kasahara). The region with any disturbance such as feeds, side streams, condenser or reboiler is provided with single tray units (models). The region of applicability of the single tray model depends entirely on the choice of the designer. Roughly one or two

single tray units above and below the disturbance are recommended but it is not a prerequisite. The remaining regions between the single trays are considered as multitray units.

3.1 Single Tray Model:

It has already been mentioned that the single tray model and the relevant equations are similar to the ~~Naphtali-Sandholm~~ method. It should, therefore, be noted that if the number of trays and the feed location are known, as in the simulation problem, all the trays can be assigned single tray models and the computer package can be executed to obtain ~~rigorous~~ solution of the rating problem with ~~Naphtali-Sandholm~~ algorithm. Figure 2 represents a generalised single tray model with the symbols having their usual significance.

The independent variables for each unit are:

- i) $l_{i,j}$ $j = 1, 2, \dots, C$
- ii) $v_{i,j}$ $j = 1, 2, \dots, C$
- iii) T_i

Here i refers to the unit number going from 1 to N and j refers to the component numbers going from 1 to C . There are a total of $(2C+1)$ independent variables associated with each single tray model. We need an equal number of equations to solve for the variables. These are:

(a) Component Material Balance Equations:

$$M_{i,j} = l_{i-1,j} - (1+s_i^L) l_{i,j} - (1+s_i^V) v_{i,j} + v_{i+1,j} + f_{i,j}^L + f_{i,j}^V \quad \text{where } j=1,2,\dots,C \quad (3.1)$$

(b) Componentwise Equilibrium Relations with Murphree Plate Efficiencies, $E_{i,j}$:

$$Q_{i,j} = \frac{E_{i,j} K_{i,j} l_{i,j}}{L_i} - \frac{v_{i,j}}{V_i} + (1-E_{i,j}) \frac{v_{i+1,j}}{V_{i+1}} \quad (3.2)$$

where $L_i = \sum_{j=1}^C l_{i,j}$ and $V_i = \sum_{j=1}^C v_{i,j}$ and $j=1,2,\dots,C$

and $K_{i,j}$ are the equilibrium constants for the j th component on the i th unit.

(c) Enthalpy balance equations:

$$E_i = \sum_{j=1}^C \left[l_{i-1,j} h_{i-1,j} + v_{i+1,j} H_{i+1,j} - (1+s_i^L) l_{i,j} h_{i,j} - (1+s_i^V) v_{i,j} H_{i,j} + f_{i,j}^L h_{i,j}^F + f_{i,j}^V H_{i,j}^F \right] + Q_i^* \quad (3.3)$$

where h denotes the enthalpy for liquid and H is the enthalpy for vapour and Q_i^* is the heat supplied or taken out from each single tray unit. The sign of Q_i^* will be positive for heat supply to the unit and negative for heat supply to the

23

unit and negative for heat withdrawal from the unit. There are $(2C+1)$ equations associated with each single tray model.

For condensers and reboilers these equations are adjusted according to the type of problem and given end specifications. The matching of the degrees of freedom with the specifications and modification of the suitable equations have been discussed in detail by Das (1985). It should be noted that the equations for the single tray model are written in terms of certain functions such as $M_{i,j}$, $Q_{i,j}$ and E_i . These are called the "discrepancy functions" and have certain values or discrepancies when the numerical iteration continues and the unknowns are not solved. These discrepancies, however, reduce to zero as convergence is reached i.e., when the solution is found.

The number of dependent variables is a minimum in this model. The dependent variables include only the equilibrium constants ($K_{i,j}$) and the enthalpy terms ($h_{i,j}$ and $H_{i,j}$), which are dependent on temperature and pressure only for the ideal case. For the non-ideal systems, comparatively more complicated functionalities arise and those will be discussed in Chapter 4.

When all the N -trays in the tower are taken as single tray units, the formulation is essentially that of Naphtali-Sandholm type and all the $N(2C+1)$ equations can be solved together to obtain the column performance. It should be noted here that all the $(2C+1)$ equations for i th unit have only three

sets of variables i.e., variables associated with (i-1)th unit, ith unit itself and (i+1)th unit. This gives rise to the symmetric tridiagonal structure in the matrix. We will proceed to discuss the matrix structures after describing the multitray model.

3.2 Multitray Model:

Figure 3 depicts the schematic representation of a general multitray unit along with the incoming and outgoing energy and mass streams. In any ith multitray unit, as shown in Figure 3, M_i trays are incorporated and this M_i can be treated, as an independent variable for design computations.

The independent variables for each multitray unit are:

- (i) $l_{i,j}$ $j = 1, 2, \dots, C$
- (ii) $v_{i,j}$ $j = 1, 2, \dots, C$
- (iii) M_i

Here i refers to the unit number going from 1 to N and j refers to the component number going from 1 to C . There are a total of $(2C+1)$ independent variables associated with each multitray model. We need an equal number of equations to solve the variables.

The equations used in the multitray model are as follows:

(a) Component Material Balance Equations:

$$M_{i,j} = v_{i+1,j} + l_{i-1,j} - (1+s_i^L) l_{i,j} - (1+s_i^V) v_{i,j} \quad (3.4)$$

where $j = 1, 2, \dots, C$

(b) Enthalpy Balance Equations:

$$E_i = \sum_{j=1}^C \left[l_{i-1,j} h_{i-1,j} - (1+s_i^L) l_{i,j} h_{i,j} - (1+s_i^V) v_{i,j} H_{i,j} \right. \\ \left. + v_{i+1,j} H_{i+1,j} + f_{i,j}^L h_{i,j}^F + f_{i,j}^V H_{i,j}^F \right] + Q_i^* \quad (3.5)$$

The material and energy balance equations are identical in case of multitray unit and single tray unit. However, the difference between the two models appears in the equilibrium equations which follow:

(c) Overall Equilibrium Equation for the Multitray Unit:

The overall equilibrium equation is obtained by condensing the platewise component material balances and equilibrium relationships and eliminating the variables inside the multitray unit as shown in the following deduction. It is known that equilibrium relation and component material balance for a general mth tray in the multitray model are given by

$$v_{i,m,j}^I + l_{i,m,j}^I = v_{i,m+1,j}^I + l_{i,m-1,j}^I \quad (3.6)$$

$$v_{i,m,j}^I = S_{i,m,j}^I l_{i,m,j}^I \quad (3.7)$$

Variables with primes indicate that they refer to trays inside the multitrays units.

Starting from one end of the multitray model, we have

$$\begin{aligned}
 v_{i,1,j}^1 &= s_{i,1,j}^1 l_{i,1,j}^1 \\
 &= s_{i,1,j}^1 (v_{i,2,j}^1 + l_{i-1,j}^1 - v_{i,1,j}^1) \\
 &\quad \text{(Using equation 3.6)} \\
 &= s_{i,1,j}^1 s_{i,2,j}^1 l_{i,2,j}^1 + s_{i,1,j}^1 (l_{i-1,j}^1 - v_{i,1,j}^1) \\
 &\quad \text{(Using equation 3.7)} \\
 &\quad \dots \dots \dots \\
 &= s_{i,1,j}^1 s_{i,2,j}^1 \dots \dots s_{i,M,j}^1 l_{i,M,j}^1 + \\
 &\quad (s_{i,1,j}^1 + s_{i,1,j}^1 s_{i,2,j}^1 + \dots \dots \dots + \\
 &\quad s_{i,1,j}^1 s_{i,2,j}^1 \dots \dots s_{i,M-1,j}^1) (l_{i-1,j}^1 - v_{i,1,j}^1)
 \end{aligned}$$

(3.8)

Let $\pi_{i,j} = s'_{i,1,j} s'_{i,2,j} \dots s'_{i,M,j}$ (3.9)

$$\text{and } D_{i,j} = S_{i,1,j}^1 + S_{i,1,j}^1 S_{i,2,j}^1 + \dots + S_{i,1,j}^1 S_{i,2,j}^1 \dots S_{i,M-1,j}^1 \quad (3.10)$$

Also, it is obvious (Figure 3) that

$$v_{i,1,j}^i = (1 + s_i^V) v_{i,j} \quad (3.11)$$

$$\text{and} \quad l_{i,M,j}^L = (1 + s_1^L) l_{i,j} \quad (3.12)$$

Ohmura and Kasahara deduced equation 2.4 by substituting equations 3.9, 3.10, 3.11 and 3.12 in equation 3.8. In the present formulation a more rigorous double stripping factor approach of Edmister has been used and a modified system equation has been formulated from which the values of the stripping factors can be directly deduced.

From equations 3.8, 3.9 and 3.10 we have

$$\begin{aligned} v'_{i,1,j} &= \pi_{i,j} l'_{i,M,j} + (D_{i,j})(l_{i-1,j} - v'_{i,1,j}) \\ &= \pi_{i,j} l'_{i,M,j} + (D_{i,j} + \pi_{i,j})(l_{i-1,j} - v'_{i,1,j}) \\ &\quad - \pi_{i,j}(l_{i-1,j} - v'_{i,1,j}) \end{aligned}$$

$$\text{or } v'_{i,1,j} (1+W_{i,j}) = \pi_{i,j} (\text{SUM}) + l_{i-1,j} (W_{i,j}) \quad (3.13)$$

$$\text{where } (\text{SUM}) = (l'_{i,M,j} - l_{i-1,j} + v'_{i,1,j}) \quad (3.14)$$

$$\text{and } W_{i,j} = (D_{i,j} + \pi_{i,j}) \quad (3.15)$$

$$\begin{aligned} \text{or } v'_{i,1,j} &= \frac{W_{i,j}}{(1+W_{i,j})} l_{i-1,j} + (\text{SUM}) \frac{\pi_{i,j}}{(1+W_{i,j})} \\ &= \frac{W_{i,j}}{(1+W_{i,j})} l_{i-1,j} + (\text{SUM}) - (\text{SUM}) \left[1 - \frac{\pi_{i,j}}{(1+W_{i,j})} \right] \\ &= (\text{SUM}) + \left[l_{i-1,j} - (\text{SUM}) \left\{ \frac{1+W_{i,j} - \pi_{i,j}}{W_{i,j}} \right\} \right] \frac{W_{i,j}}{(1+W_{i,j})} \\ \text{or } \frac{v'_{i,1,j} - (\text{SUM})}{l_{i-1,j}} &= \left[1 - \frac{(\text{SUM})}{l_{i-1,j} S'_{i,j}} \right] \frac{W_{i,j}}{(1+W_{i,j})} \quad (3.16) \end{aligned}$$

Using eqn. 3.14 in eqn. 3.16 we have

$$\frac{l_{i-1,j} - l'_{M_{i,j}}}{l_{i-1,j}} = \left[1 - \frac{(\text{SUM})}{l_{i-1,j} S'_{i,j}} \right] \left(\frac{W_{i,j}}{1+W_{i,j}} \right) \quad (3.17)$$

where $S'_{i,j} = \frac{W_{i,j}}{1+W_{i,j} - \pi_{i,j}}$ (3.18)

Let $Se_{i,j}$ represent the effective stripping factor for the j th component in the i th unit.

Therefore: $\pi_{i,j} = Se_{i,j}^{M_{i,j}}$ (3.19)

and $D_{i,j} = \frac{(Se_{i,j}^{M_{i,j}} - Se_{i,j})}{(Se_{i,j} - 1)}$ (3.20)

and $W_{i,j} = Se_{i,j} + Se_{i,j}^2 + \dots + Se_{i,j}^{M_{i,j}}$
 $= (Se_{i,j}^{M_{i,j}+1} - Se_{i,j}) / (Se_{i,j} - 1)$ (3.21)

and $(W_{i,j} + 1) = Se_{i,j} + Se_{i,j}^2 + \dots + Se_{i,j}^{M_{i,j}} + 1$
 $= (Se_{i,j}^{M_{i,j}+1} - 1) / (Se_{i,j} - 1)$ (3.22)

Now substituting equations 3.19 to 3.22 in eqns. 3.17 and 3.18 we have:

$$\frac{l_{i-1,j} - l'_{M_{i,j}}}{l_{i-1,j}} = \left[1 - \frac{(\text{SUM})}{l_{i-1,j} S'_{i,j}} \right] \frac{Se_{i,j}^{M_{i,j}+1} - Se_{i,j}}{Se_{i,j}^{M_{i,j}+1} - 1} \quad (3.23)$$

Edmister (1943) in his deduction used the assumption that most of the mass transfer takes place in the terminal plates of the column only and hence the expressions for the effectiveness stripping factors can be deduced using a column with two trays (i.e., by taking the first and Mth trays only)

So,

$$\frac{S_{e,i,j}^3 - S_{e,i,j}}{S_{e,i,j}^3 - 1} = \frac{S_{i,1,j}^! + S_{i,1,j}^! S_{i,M,j}^!}{(S_{i,1,j}^! + S_{i,1,j}^! S_{i,M,j}^{!+1})} \quad (3.24)$$

By factoring out the term $(S_{e,i,j} - 1)$ and simplifying we have the quadratic equation

$$S_{e,i,j}^2 + S_{e,i,j} - S_{i,1,j}^! (1 + S_{i,M,j}^!) = 0$$

which has the positive solution

$$S_{e,i,j} = [S_{i,1,j}^! (1 + S_{i,M,j}^!) + 0.25]^{0.5} - 0.5 \quad (3.25)$$

Similarly, from equations 3.18 we have

$$\begin{aligned} S_{i,j}^! &= \frac{W_{i,j}}{1 + W_{i,j} - \bar{W}_{i,j}} = \frac{S_{i,1,j}^! + S_{i,1,j}^! S_{i,M,j}^!}{1 + S_{i,1,j}^! + S_{i,1,j}^! S_{i,M,j}^! - (S_{i,1,j}^! * S_{i,M,j}^!)} \\ &= \frac{S_{i,1,j}^! (1 + S_{i,M,j}^!)}{(1 + S_{i,1,j}^!)} \quad (3.26) \end{aligned}$$

$S_{i,j}^!$ is the second stripping factor term of Edmister.

Now substituting equations: 3.11 and 3.12 in equation 3.23,

we have:

$$\frac{l_{i-1,j} - (1+s_i^L) l_{i,j}}{l_{i-1,j}} = \left[1 - \frac{(\text{SUM})}{l_{i-1,j} S_{i,j}^L} \right] \frac{Se_{i,j}^{M_i+1} - Se_{i,j}}{Se_{i,j}^{M_i+1} - 1} \quad (3.27)$$

$$\text{where } (\text{SUM}) = (1+s_i^L) l_{i,j} - l_{i-1,j} + (1+s_i^V) v_{i,j} \quad (3.28)$$

This is the rigorous form of the overall equilibrium equation to be used for the multitray units in the convergence calculations.

It should be noted here that the expressions for the effective stripping factors consist of two terms $S_{i,1,j}^L$ and $S_{i,M,j}^L$ which are given by:

$$S_{i,1,j}^L = (K_{i,1,j}^L v_i (1+s_i^V)) / (L_{i,1}^L) \quad (3.29)$$

$$\text{and } S_{i,M,j}^L = (K_{i,M,j}^L v_{i,M}^L) / (L_i (1+s_i^L)) \quad (3.30)$$

and here we have two dependent variables $L_{i,1}^L$ and $v_{i,M}^L$ which have not been defined so far. These variables can be associated in two ways.

(i) Approximate method: In this approach, the flow profile inside the multitray model is assumed to be linear. This necessarily implies that the flowrates on successive trays increase or decrease by a constant value throughout the multitray model. That is, we have

$$L_{i,1}^I = \frac{L_{i-1} - L_i (1 + s_i^L)}{M_i} + L_{i-1} \quad (3.31)$$

$$V_{i,M}^I = \frac{V_i (1 + s_i^V) - V_{i+1}}{M_i} + V_{i+1} \quad (3.32)$$

For very simple problems, this method can be used to simplify computational efforts and thereby save time. However, only an approximate solution is obtained in this case and the convergence cannot be guaranteed. For complex, nonideal cases or problems with irregular flow profiles the rigorous approach, to be discussed, is suggested.

(ii) Rigorous Method: In this method, $L_{i,1}^I$ and $V_{i,M}^I$ are calculated rigorously from the system equations using mathematical techniques.

The general dew point and bubble point equations for the components leaving the i th unit are given by

$$\frac{1}{V_i} \sum_{j=1}^c \frac{v_{i,j}}{K_{i,j}} - 1 = 0 = f(T_i) \quad (3.33)$$

and

$$\frac{1}{L_i} \sum_{j=1}^c l_{i,j} K_{i,j} - 1 = 0 = f(T_i) \quad (3.34)$$

Now, for the multitray model, considering the Murphree tray efficiencies to be unity, the vapour leaving from the top is at the dew point temperature, $T_{i,1}^I$ and the liquid leaving from the bottom is at its bubble point temperature, $T_{i,M}^I$. If the component vapour and liquid rates leaving the multitray unit

are known, $T_{i,1}$ and $T_{i,M}$ can be calculated from equations (3.33) and 3.34 respectively. Consequently, the mole fractions of the liquid leaving the first tray in the multitray model (i.e. $x_{i,j}^I$) can be obtained from:

$$\frac{v_{i,j}}{v_i} = K_{i,1,j}^I \quad x_{i,1,j}^I = 0 \quad (j=1,2,\dots,C) \quad (3.35)$$

The enthalpy balance for the first tray inside the multitray model is given by

$$\sum_{j=1}^C \left[v_{i,1,j}^I H_{i,1,j}^I + l_{i,1,j}^I h_{i,1,j}^I - l_{i-1,j}^I h_{i-1,j}^I - v_{i,2,j}^I H_{i,2,j}^I \right] = 0 \quad (3.36)$$

Also,

$$l_{i,1,j}^I = x_{i,1,j}^I L_{i,1}^I \quad (3.37)$$

$$\text{and} \quad v_{i,2,j}^I = v_{i,1,j}^I + l_{i,1,j}^I - l_{i-1,j}^I \quad (3.38)$$

Therefore, substituting equations 3.37 and 3.38 in equation 3.36 we have

$$\sum_{j=1}^C \left[(1+s_i^V) v_{i,j}^I H_{i,1,j}^I + x_{i,1,j}^I L_{i,1}^I h_{i,1,j}^I - l_{i-1,j}^I h_{i-1,j}^I - \left\{ (1+s_i^V) v_{i,j}^I + x_{i,1,j}^I L_{i,1}^I - l_{i-1,j}^I \right\} H_{i,2,j}^I \right] = 0 \quad (3.39)$$

$H'_{2,i,j}$, the enthalpy term for vapour leaving the second tray inside the multitray unit, can be computed if the plate temperature $T'_{2,i}$ is known, which, in turn, can be calculated from an equation (Eqn. 3.40) similar to Eqn. 3.33, since $v'_{i,2,j}$ are known as functions of $L'_{i,1}$.

$$\frac{1}{v'_{i,2}} \sum_{j=1}^C \frac{v'_{i,2,j}}{K'_{i,2,j}} - 1 = 0 = f(T'_{i,2}) \quad (3.40)$$

Hence the nonlinear equation 3.39 can be written in the form $f(L'_{i,1}) = 0$ from which $L'_{i,1}$ can be numerically solved using Regula-falsi technique. Similarly, by using the component material balance, equilibrium and enthalpy balance for the bottom plate of the multitray model, $v'_{i,M}$ can be computed. The relevant equations are as follows.

Enthalpy balance for the bottom tray inside the multitray model is given by

$$\sum_{j=1}^C [v'_{i,M,j} H'_{i,M,j} + l'_{i,M,j} h'_{i,M,j} - l'_{i,M-1,j} h'_{i,M-1,j} - v'_{i+1,j} H'_{i+1,j}] = 0 \quad (3.41)$$

where

$$v'_{i,M,j} = y'_{i,M,j} v'_{i,M} \quad (3.42)$$

$$\text{and } l'_{i,M-1,j} = (v'_{i,M,j} + l'_{i,M,j} - v'_{i+1,j}) \quad (3.43)$$

$$\text{with } \frac{1}{L_1} \sum_{j=1}^c K'_{i,M,j} l_{i,j}^{-1} = 0 = f(T'_{i,M}) \quad (3.44)$$

$$\text{and } \frac{1}{L'_{i,M-1}} \sum_{j=1}^c K'_{i,M-1,j} l'_{i,M-1,j}^{-1} = 0 = f(T'_{i,M-1}) \quad (3.45)$$

3.3 Variables Selection and Matrix Structure:

It has already been shown that each single tray unit has $(2C+1)$ independent variables and equal number of equations. Retaining the consistency of size and symmetry in structure of the Jacobian matrix for all the units in the column is important for CPU time and memory reduction. This essentially demands that the number of independent variables be restricted to $(2C+1)$ only. The output variables, i.e., vapour and liquid flow rates issuing out of the multitray units already constitute $2C$ independent variables.

Further, the multitray units have two terminal temperatures to be considered per unit in addition to the number of trays, M_i . If all these are treated as independent variables, the total number for multitray units will be $2C+3$. This will destroy the constancy of size in the Jacobian matrix, the number of variables for a single tray unit being $2C+1$. To overcome this problem, only the number of trays, M_i , has been taken to be an independent variable. The terminal temperatures along with the total vapour and liquid rates discussed earlier have been considered as dependent variables and are computed, whenever required, from

the set of independent variables. To sum up, following have been selected as independent and dependent variables for multitray unit:

A. Independent Variables

Variable Name Number

(a) component vapour flow rates
from the top of the unit

$v_{i,j}$ C

(b) Component liquid flow rates
from the bottom of the unit

$l_{i,j}$ C

(c) Number of plates inside the
unit

M_i 1

The number of independent variables is $2C+1$ per unit which is equal to the number of independent discrepancies specified for the unit.

B. Dependent Variables:

(a) Total vapour flow rate from
the top plate of the unit

$$V_i = \sum_{j=1}^C v_{i,j}$$

(b) Total liquid flow rate from
the bottom plate of the unit

$$L_i = \sum_{j=1}^C l_{i,j}$$

(c) Top plate temperature of the unit

$T_{i,1}$ evaluated from
the dew point equation
of the outgoing
vapour

- | | |
|---|---|
| (d) Bottom plate temperature
of the unit | $T_{i,M}^1$ evaluated from the
bubble point equation
of the outgoing liquid |
| (e) Top plate total liquid
flow rate | $L_{i,1}^1$ evaluated as mentioned
in Section 3.2. |
| (f) Bottom plate total vapour
flow rate | $V_{i,M}^1$ evaluated as mentioned
in Section 3.2 |

There are in total, $N(2C+1)$ independent variables and an equal number of equations where N is the total number of units in the column. Now we proceed to introduce the vector notations for the column matrices. Consider \bar{X}_i as the vector of variables for unit i and \bar{F}_i as vector of functions for the same unit.

$$\bar{X} = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_N \end{bmatrix} \quad \text{and} \quad \bar{F} = \begin{bmatrix} \bar{F}_1 \\ \bar{F}_2 \\ \vdots \\ \bar{F}_N \end{bmatrix} \quad (3.46)$$

where,

$$\bar{X}_i = \begin{bmatrix} v_{i,1} \\ v_{i,2} \\ \vdots \\ v_{i,C} \\ l_{i,1} \\ l_{i,2} \\ \vdots \\ l_{i,C} \\ T_i(\text{or } M_i) \end{bmatrix} = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,C} \\ x_{i,C+1} \\ x_{i,C+2} \\ \vdots \\ x_{i,2C} \\ x_{i,2C+1} \end{bmatrix}; \bar{F}_i = \begin{bmatrix} M_{i,1} \\ M_{i,2} \\ \vdots \\ M_{i,C} \\ Q_{i,1} \\ Q_{i,2} \\ \vdots \\ Q_{i,C} \\ E_i \end{bmatrix} = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,C} \\ f_{i,C+1} \\ f_{i,C+2} \\ \vdots \\ f_{i,2C} \\ f_{i,2C+1} \end{bmatrix} \quad (3.47)$$

\bar{J} , the Jacobian matrix which is really a matrix of the partial derivatives of all the functions with respect to all the variables at the present value of the \bar{X}^I , is given by

$$\bar{J} = \left[\frac{\partial \bar{F}}{\partial X} \right] = \begin{bmatrix} \left[\frac{\partial \bar{F}_1}{\partial x_1} \right] & \left[\frac{\partial \bar{F}_1}{\partial x_2} \right] & \dots & \left[\frac{\partial \bar{F}_1}{\partial x_N} \right] \\ \left[\frac{\partial \bar{F}_2}{\partial x_1} \right] & \left[\frac{\partial \bar{F}_2}{\partial x_2} \right] & \dots & \left[\frac{\partial \bar{F}_2}{\partial x_N} \right] \\ \vdots & \vdots & \ddots & \vdots \\ \left[\frac{\partial \bar{F}_N}{\partial x_1} \right] & \left[\frac{\partial \bar{F}_N}{\partial x_2} \right] & \dots & \left[\frac{\partial \bar{F}_N}{\partial x_N} \right] \end{bmatrix} \quad (3.48)$$

with

$$\left[\frac{\partial \bar{F}_i}{\partial x_j} \right] = \begin{bmatrix} \frac{\partial F_{i,1}}{\partial x_{j,1}} & \frac{\partial F_{i,1}}{\partial x_{j,2}} & \dots & \frac{\partial F_{i,1}}{\partial x_{j,2C+1}} \\ \vdots & & & \\ \frac{\partial F_{i,2C+1}}{\partial x_{j,1}} & \dots & \dots & \frac{\partial F_{i,2C+1}}{\partial x_{j,2C+1}} \end{bmatrix} \quad (3.49)$$

Fortunately, the distillation column equations contain variables pertaining to three consecutive plates, namely, $(i-1), i$ and $(i+1)$ only and therefore, in the above matrix most partial derivative terms are zero except three and hence the Jacobian matrix assumes the block tri-diagonal form as follows:

$$\bar{J} = \left[\frac{\partial \bar{F}}{\partial X} \right] = \begin{bmatrix} \bar{B}_1 & \bar{C}_1 & 0 & 0 & \dots & 0 \\ \bar{A}_2 & \bar{B}_2 & \bar{C}_2 & 0 & 0 & 0 \\ 0 & \bar{A}_3 & \bar{B}_3 & \bar{C}_3 & 0 & 0 \\ 0 & & & & & \\ 0 & 0 & 0 & & \bar{A}_N & \bar{B}_N \end{bmatrix} \quad (3.50)$$

$$\text{where } \bar{A}_i = \left[\frac{\partial \bar{F}_i}{\partial x_{i-1}} \right]; \quad \bar{B}_i = \left[\frac{\partial \bar{F}_i}{\partial x_i} \right]; \quad \bar{C}_i = \left[\frac{\partial \bar{F}_i}{\partial x_{i+1}} \right] \quad (3.51)$$

Now we proceed to determine the structures of the submatrices \bar{A}_i , \bar{B}_i and \bar{C}_i in the overall Jacobian matrix. The structure for the single tray model will be similar to the Naphtali-Sandholm algorithm and is shown in Figure 4. For the multitray models, however, there will be a change in the structural pattern as shown in Figure 5. The cross marks (X) in the pattern indicate the presence of the derivative term at that location. However, this partial derivative can be calculated either numerically, analytically or in a mixed mode so as to minimize computer memory and CPU time requirements.

3.4 Solution Procedure:

Newton-Raphson solution procedure for a set of nonlinear equations requires

$$\bar{x}^{I+1} = \bar{x}^I + \Delta \bar{x}^I \quad (3.52)$$

where I and I+1 are iteration numbers and $\Delta \bar{x}$ is the correction vector, given by

$$\Delta \bar{x}^I = - \left[\bar{J}^{-1} \right]^I \bar{F}^I \quad (3.53)$$

In order to evaluate the correction vector, one must compute inverse of the Jacobian matrix \bar{J}^{-1} . Of the various procedures available, Gaussian Elimination and Thomas Algorithm are the most important. Since the structure is block tridiagonal the Block-Thomas Algorithm is the most efficient. The user has been provided with the option to choose either of the two

procedures in the present program (Appendix F). A detailed discussion on these algorithms and their CPU time and memory saving capabilities has been provided in a companion report (Das, 1985).

However, it is not guaranteed that the correction vector, when used as such, will ensure convergence. There can be a number of reasons which can lead to a diverging or oscillating behaviour. For example, the convergence is sensitive to initial guesses and the local behaviour of the Jacobian matrix. Therefore, it is suggested that a fraction of the correction vector is taken and equation 3.52 should be rewritten as

$$\bar{x}^{I+1} = \bar{x}^I + \beta \Delta \bar{x}^I \quad (3.54)$$

where β lies between -1 and 1 (Kaufmann, 1983). To implement equation 3.54, it is absolutely necessary that β is selected in such a way that convergence will be ensured. To achieve this criterion, a single dimensional Golden Section (Beveridge and Schechter, 1970) search routine has been incorporated.

The unidimensional numerical optimization for unimodal functions using Golden Section technique involves the following steps.

Step 1: Define the range of search for β i.e., $\beta_1 \leq \beta \leq \beta_f$

Step 2: Locate two points (β_1 and β_2) at a distance of

$(1-l)$ and l from one end of the total range, β_1

where $l = \frac{\sqrt{5}-1}{2} = 0.6180$ (approx.)

Step 3: Calculate the variables \bar{X}^{I+1} from equation: 3.48

for $\beta = \beta_1$

Step 4: Repeat Step 3 for $\beta = \beta_2$

Step 5: Calculate the overall sum of the squares of all discrepancies, $(SUM)_1$ and $(SUM)_2$ for $\beta = \beta_1$ and $\beta = \beta_2$ respectively.

Step 6: Calculate $(SUM)_1$ and $(SUM)_f$ corresponding to the value of the variables \bar{X}^{I+1} from equation: 3.48 with $\beta = \beta_1$ and $\beta = \beta_f$ respectively.

Step 7: A new interval of search is selected, thus:

(a) If $(SUM)_1$ equals $(SUM)_2$ the new interval for β becomes $\beta_1 \leq \beta \leq \beta_2$

(b) If $(SUM)_1$ is greater than $(SUM)_2$, the new interval for β becomes $\beta_1 \leq \beta \leq \beta_f$.

(c) If $(SUM)_1$ is less than $(SUM)_2$, the new interval for β becomes $\beta_i \leq \beta \leq \beta_2$.

Step 8: Define appropriately β_1 and β_f , the boundaries of of the new range of search for β , according to the particular case and repeat Step-2 to Step-7 till the absolute value of $(\beta_1 - \beta_2)$ is less than or equal to ϵ where ϵ is the supplied tolerance limit.

It should be noted that for most of the ideal cases the value of β tends to unity. However, for complex situations β assumes different optimal values to guide the iterations towards convergence.

	Variable Numbers								
	1	2	...	C	C+1	C+2	...	2C	2C+1
Function numbers									
1	x	0	...	0	0	0	...	0	0
2	0	x	...	0	0	0	...	0	0
...
C	0	0	...	x	0	0	...	0	0
C+1	0	0	...	0	0	0	...	0	0
...
2C	0	0	...	0	0	0	...	0	0
2C+1	x	x	...	x	0	0	...	0	x

Figure 4(a): Submatrix \bar{A}_1

	:Variable numbers								
	1	2	...	C	C+1	C+2	...	2C	2C+1
1	x	0	...	0	x	0	...	0	0
2	0	x	...	0	0	x	...	0	0
...
C	0	0	...	x	0	0	...	x	0
C+1	x	x	...	x	x	x	...	x	x
C+2	x	x	...	x	x	x	...	x	x
...
2C	x	x	...	x	x	x	...	x	x
2C+1	x	x	...	x	x	x	...	x	x

Figure: 4(b) Submatrix \bar{B}_1

$$C \equiv I =$$

	: Variable numbers									
	1	2	...	C	C+1	C+2	...	2C	2C+1	
Function numbers	1	0	0	...	0	x	0	...	0	0
	2	0	0	...	0	0	x	...	0	0

	C	0	0	...	0	0	0	...	x	0
	C+1	0	0	...	0	x	x	...	x	0
	C+2	0	0	...	0	x	x	...	x	0

	2C	0	0	...	0	x	x	...	x	0
	2C+1	0	0	...	0	x	x	...	x	x

Figure 4(c): Submatrix \bar{C}_1

Figure 4: Structural pattern in the submatrices A_1 , B_1 and C_1 of the overall Jacobian matrix where i represents a single tray unit number only.

		:Variable numbers:								
		1	2	...	C	C+1	C+2	...	2C	2C+1
$\bar{A}_1 =$	Function numbers									
	1	x	0	...	0	0	0	...	0	0
	2	0	x	...	0	0	0	...	0	0

	C	0	0	...	x	0	0	...	0	0
	C+1	x	x	...	x	0	0	...	0	x
	C+2	x	x	...	x	0	0	...	0	x

	2C	x	x	...	x	0	0	...	0	x
	2C+1	x	x	...	x	0	0	...	0	x

Figure 5(a): Submatrix \bar{A}_1

		:Variable numbers:								
		1	2	...	C	C+1	C+2	...	2C	2C+1
Function numbers \bar{B}_1	1	x	0	...	0	x	0	...	0	0
	2	0	x	...	0	0	x	...	0	0

	C	0	0	...	x	0	0	...	x	0
	C+1	x	x	...	x	x	x	...	x	x
	C+2	x	x	...	x	x	x	...	x	x

	2C	x	x	...	x	x	x	...	x	x
2C+1	x	x	...	x	x	x	...	x	0	

Figure 5(b); Submatrix \bar{B}_1

	:Variable numbers:								
	1	2	...	c	C+1	C+2	...	2C	2C+1
1	0	0	...	0	x	0	...	0	0
2	0	0	...	0	0	x	...	0	0
...
C	0	0	...	0	0	0	...	x	0
C+1	0	0	...	0	x	x	...	x	x
C+2	0	0	...	0	x	x	...	x	x
...
2C	0	0	...	0	x	x	...	x	x
2C+1	0	0	...	0	x	x	...	x	x

Figure 5(c): Submatrix \bar{C}_i

Figure 5: Structural pattern in the submatrices A_i , B_i and C_i of the overall Jacobian matrix where i represents a multitray unit number only

CHAPTER 4

THERMODYNAMIC PROPERTY ESTIMATIONS FOR MULTICOMPONENT DISTILLATION CALCULATIONS

4.1 Introduction:

A prerequisite for any simulation (or design) calculation of a chemical plant or process equipment is the knowledge of physical and thermodynamic data of various streams (pure components and their mixtures) in the process. Almost all good simulator packages contain an internal library of thermo-physical data and also have inbuilt programmes for the estimation of thermodynamic and physical properties.

To solve the distillation problems involving multi-component mixtures, the vapour-liquid equilibrium data and enthalpy data are required. The methods used to obtain these data may be classified as follows:

1. Curve fits of pure component data obtained experimentally or from generalized correlations
2. Generalized correlations as functions of temperature, pressure and composition for both pure components and mixtures.
3. Techniques using a single equation of state for both liquid and vapour phases. These are known as "Direct Methods".

4. Procedures using an equation of state or correlation for the vapour phase and a group contribution method for the liquid phase. These are known as "Indirect Methods".

The first category methods are used when a large amount of experimental data is available. This is usually recommended for very odd cases having association and other interactions or for uncommon systems with unestablished thermodynamic behaviour. The methods of the second category utilise some correlations, which have been repeatedly tested and validated for a large number of systems. These are extremely useful for ideal mixtures, however, they often pose problems when dealing with nonideal systems. Expressions, with respect to temperature, pressure and vapour and liquid phase mole fractions, having generalized applicability to all systems at all ranges of conditions, are very difficult to obtain. Examples of these methods are Antoine Equation (1888); Yen and Alexander (1965) correlations etc. The third category of methods emerged recently when it was found that the same equation of state can be applied to both liquid and vapor phases in most of the common systems with fair degree of accuracy. However, the most commonly used equations of states sometimes fail to predict the liquid phase nonidealities of highly associated or interacting liquids. The industrial uses of the equations of state

have been elaborated in paper by Adler et al. (1977).

For complicated industrial systems, the use of multiple equations of state for liquid and vapor phases has been suggested by Hollan (1981). These equations have been elaborated in Section 4.3.

For systems having highly interacting or associated behaviour, a single equation of state often fails to predict the behaviour in both liquid and vapour phases. In such cases, the fourth category of methods i.e., the indirect methods is recommended. It utilizes molecular interaction or group interaction parameters to predict the liquid phase activity coefficients. This method is particularly useful to associated systems like alcohols and will be presented more elaborately in Section 4.3. However, it should be mentioned that the degree of accuracy required and the exact method to be used is upto the choice of the designer and a trade off between accuracy and computer timing should be made in accordance to the requirements.

4.2. Basic Property Estimation for Pure Components:

The basic properties which are required as an information feed to the rigorous nonideal calculations, include the physical constants like molecular weights, boiling points and critical properties for pure components. For very common compounds it is suggested that these values can either be directly given as an input or may be automatically read from a property data bank. However, for uncommon systems like halosubstituted

silanes very little information is available in literature. The reported experimental data for such uncommon systems has an uncertainty as high as 40% to 60%. For such cases estimation of properties from the molecular weight and boiling point data is essential.

4.2.1 Critical Property Estimation:

Critical temperature, pressure and volume (i.e. T_c , P_c and V_c) represent three widely used pure component constants. The critical properties for most of the common compounds have been given by Reid et al. (1977). Comprehensive reviews of critical properties are available by Kudchadkar et al. (1968) for organic compounds and Mathews (1972) for inorganic compounds. When estimation is required, the method of Lydersen (1955) is normally employed. In this method, structural contributions for temperature, pressure and volume (i.e. Δ_T , Δ_P and Δ_V respectively) are employed to estimate T_c , P_c and V_c . The relations are

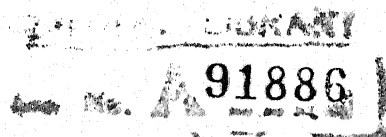
$$T_c = T_b \left[0.567 + \sum \Delta_T - (\sum \Delta_T)^2 \right]^{-1} \quad (4.1)$$

$$P_c = M (0.34 + \sum \Delta_P)^{-2} \quad (4.2)$$

$$\text{and } V_c = (40 + \sum \Delta_V) \quad (4.3)$$

where T_b = normal boiling point of the compound in degrees Kelvin

and M = molecular weight of the compound.



Values of Δ_T , Δ_P and Δ_V for various compounds have been given by Reid et al. (1977) as well as by Lydersen (1955). The T_c , P_c and V_c are obtained in degrees Kelvin, atmospheres and cubic centimeters per gram mole respectively. The errors associated in estimation are usually less than 2% for T_c and less than 4% for P_c . Δ_V data for all compounds are not available for the Lydersen's method. So for some organic compounds the method of Riedel (1954) is quite accurate. In this method

$$V_c = \frac{R T_c}{P_c} \left[3.72 + 0.26 (\alpha_c - 7.0) \right]^{-1} \quad (4.4)$$

where

$$\alpha_c = 0.9076 \left[1.0 + \frac{(T_b/T_c) \ln P_c}{1.0 - T_b/T_c} \right] \quad (4.5)$$

with P_c in atmospheres. R is the universal gas constant, $82.06 \text{ cm}^3 \text{ atm/gmmole K}$

α_c is known as the "Riedel factor".

At the critical point, the critical compressibility factor, Z_c , is given by

$$Z_c = \frac{P_c V_c}{RT_c} \quad (4.6)$$

The estimated critical properties for the halogen-substituted silane system have been shown in Table 2 as an example.

4.2.2 Estimation of Acentric Factor:

One of the most common pure component constants is the Pitzer acentric factor, (Pitzer, 1955), which is defined as

$$\omega = -\log P_{VP,r} \text{ (at } T_r = 0.7) - 1.0 \quad (4.7)$$

where $T_r = T_c/T_b$ and $\quad (4.8)$

$$P_{VP,r} = P_{VP}/P_c \text{ (} P_{VP} \text{ is the vapour pressure) } \quad (4.9)$$

As originally proposed, ω was to represent only the acentricity or nonsphericity of a molecule. For monatomic gases, ω is, therefore, essentially zero. For higher molecular weight hydrocarbons, ω increases and often rises with polarity. At present it is widely used as a parameter, which in some manner, measures the complexity of a molecule with respect to both the geometry and polarity. It should be noted that for strongly polar or hydrogen bonded-fluids, the above correlation is liable to involve errors.

To estimate the acentric factor by equation 4.7, the term P_{VP} is required, which, in turn, depends upon the vapour pressure equation used.

Edmister (1958) suggested

$$\log P_{VP} = (A + \frac{B}{T}) \text{ where } A \text{ and } B \text{ are constants}$$

and $\omega = \frac{3}{7} \left[\frac{\theta}{1-\theta} \right] \log P_c - 1 \quad (4.10)$

with $\theta = T_b/T_c \quad (4.11)$

Lee and Kesler (1975) suggested an improved correlation for ω . The relation is

$$\omega = \frac{-\ln P_c - 5.92714 + 6.09648 \theta^{-1} + 1.28862 \ln \theta - 0.169347 \theta^6}{15.2518 - 15.6875 \theta^{-1} - 13.4721 \ln \theta + 0.43577 \theta^6} \quad (4.12)$$

The estimated acentric factors for the halosubstituted silines systems have been reported in Table 2.

4.3 Calculation of Partition Coefficients, $K_{i,j}$:

In distillation calculations, the nonideality of thermodynamics enter mainly through the partition coefficients or equilibrium constants for the i th stage and j th component, $K_{i,j}$. If the system to be studied is ideal or near ideal, the $K_{i,j}$ can be approximated to be a function of temperature T_i and pressure P only. However, if the system is nonideal, the $K_{i,j}$'s apart from the above functionality, becomes a function of the liquid and vapour molefractions ($x_{i,j}$ and $y_{i,j}$) also. For convenience sake, in this chapter, we will drop the subscript i and denote the equilibrium constant as K_j . Now we proceed to the estimation procedures.

4.3.1 Estimation of K_j 's for Ideal Systems:

As already mentioned, for ideal systems

$$K_j = f(T, P) \quad (4.13)$$

For such cases, either direct curve fit expressions (Owens and Maddox, 1968) or vapour pressure temperature correlations are used. The most widely used expression for vapour pressure in the Antoine Equation, which is of the form

$$\ln (\text{vapour pressure } P_{VP})_j = A_j - \frac{B_j}{T + C_j} \quad (4.14)$$

where $P_{VP j}$ is in mm of mercury and

T is in $^{\circ}\text{C}$

A_j , B_j and C_j are known as Antoine Equation constants and are characteristic of the particular compound under consideration. Extensive tabulation of these constants is available in Reid et al. (1977). The partition coefficient K_j can be calculated from

$$K_j = \frac{P_{VP j}}{P} \quad (4.15)$$

where P is the system pressure in mm of mercury.

4.3.2 Estimation of K_j 's Using Direct Methods:

The criteria that must be satisfied for equilibrium of a liquid and a vapour phase at temperature T and pressure P is

$$f_j^V = f_j^L \quad j = 1, 2, \dots, C \quad (4.16)$$

where f_j is the fugacity of component j in the vapour (V) phase or liquid (L) phase and C is the number of components in the mixture. For a mixture where the vapour phase behaves like an ideal gas, one may write

$$f_j^V = y_j P \quad (4.17)$$

where y_j is the mole fraction of component j in the vapour phase. The nonideality in the vapour phase, if any, can be

taken into account by defining the fugacity coefficient ϕ_j^V as follows

$$f_j^V = \phi_j^V y_j P \quad (4.18)$$

Except for mixtures containing strongly associating substances such as organic acids, the value of ϕ_j^V will be close to unity at low pressures.

Similarly, the nonideality of the liquid phase can be accounted for by considering a liquid phase fugacity coefficient, ϕ_j^L as follows:

$$f_j^L = \phi_j^L x_j P \quad (4.19)$$

where x_j is the mole fraction of component j in the liquid phase.

From equation 4.16, at equilibrium

$$\phi_j^V y_j P = \phi_j^L x_j P$$

$$\text{or} \quad K_j = \frac{y_j}{x_j} \text{ (by definition) } = \frac{\phi_j^L}{\phi_j^V} \quad (4.20)$$

The above procedure for calculating partition coefficients has the limitation that a single equation of state is required which is valid for both the liquid and vapour phases for the whole range of temperature and pressures. For this purpose, as high as eight parameter equations of state have been proposed (e.g., the Benedict - Webb - Rubin equation of state). However, such procedures are highly uneconomic with respect to computation.

time and memory requirements specially for distillation calculations. Some of the three parameter equations of state (SRK or Peng-Robinson) have been reported to give good results, particularly for hydrocarbon system and petroleum fractions.

For the calculation of partition coefficients by this method, the equation of state is used to calculate the ϕ_j^V and ϕ_j^L and the partition coefficient is calculated from equation 4.20. The Soave modification of the Redlich-Kwong equation (SRK) method and the Peng-Robinson equation have been presented in Table 3 and 4 in different forms applicable to pure components and mixtures.

It is convenient to rearrange Equation 4.46 by multiplying and dividing the terms on the RHS by P/RT and simplifying to the following form

$$Z^3 - Z^2 + Z(A - B - B^2) - AB = 0 \quad (4.21)$$

where $Z = \text{compressibility factor} = \frac{Pv}{RT}$

$$A = \frac{\Omega_a}{\Omega_b} BF \quad (4.22)$$

$$B = \sum_{j=1}^c Z_j B_j \quad (4.23)$$

$$\text{and } B_j = \frac{b_j P}{RT} \quad (4.24)$$

Equation 4.21 has three roots and the success of the single equation of state method is due to the fact that the highest real root can be taken to correspond to the vapour phase and the lowest real root to the liquid phase. The method for finding the cubic roots is given in Appendix E.

Once Z has been calculated, v can be found from

$$v = \frac{ZRT}{P} \quad (4.25)$$

The fugacity coefficients required to calculate K_j 's for both the liquid and the vapour phases can be evaluated by using the following derivation. If μ_j and n_j represent the chemical potential and number of moles of the j th component, we know

$$\left[\left(\frac{\partial A}{\partial n_j} \right)_{T, V, n_k \neq j} \right] = \mu_j \quad (4.26)$$

$$\therefore \mu_j - \mu_j^0 = \left[\frac{\partial}{\partial n_j} (A - A^0) \right]_{T, V, n_k \neq j} \quad (4.27)$$

Again, by definition for mixtures:

$$d\mu_j = RT d \ln f_j \quad (\text{at constant } T) \quad (4.28)$$

Integrating Eqn. 4.28,

$$\mu_j - \mu_j^0 = RT \ln \frac{f_j}{f_j^0} \quad (4.29)$$

Choosing the reference state to be ideal, we have

$$f_j^0 = y_j P$$

Combining equations 4.27 and 4.29, we have

$$RT \ln \left(\frac{f_j}{f_j^o} \right) = \left[\frac{\partial}{\partial n_j} (A - A^o) \right]_{T, V, n_K \neq j} \quad (4.30)$$

or

$$RT \ln \frac{f_j}{Y_i^P} = RT \ln \phi_j = \left[\frac{\partial}{\partial n_j} (A - A^o) \right]_{T, V, n_K \neq j} \quad (4.31)$$

Again, if A is the Helmholtz free energy, we know that

$$dA = -PdV$$

or $A - A^o = - \int_{V_o}^V P dV \quad (4.32)$

Now using equation 4.42 and the SRK equation of state and then using the result in equation 4.31, finally, we have

$$\ln \phi_j = \frac{b_j}{b} (Z-1) + \ln \frac{v}{Z(v-b)} + \left(\frac{b}{bRT} \right) \ln \left(\frac{v+b}{v} \right) ** (TERM) \quad (4.33)$$

where

$$TERM = \left[\frac{b_j}{b} - 2 \sum_{k=1}^c (1 - K_{jk}) \frac{(a_j a_k)^{1/2}}{a} Z_k \right] \quad (4.34)$$

with

Z_K = liquid phase mole fraction to give ϕ_j^L and the vapour phase mole fraction to give ϕ_j^V

$$a_j = \frac{a}{b} RT b_j F_j \quad (4.35)$$

$$a = \frac{\Omega_a}{\Omega_b} \quad RT \quad bF \quad (4.36)$$

A similar procedure can be followed to calculate fugacity coefficients using Peng-Robinson equation of state in place of the SRK equation.

The partition coefficient can thus be calculated by using equation 4.20.

The results obtained for nearly ideal and nonideal systems have been shown in Chapter 5 (Problems 1 and 2).

4.3.3 Estimation of K_j 's Using Indirect Methods:

In the indirect methods, the liquid phase nonideality is characterized by the activity coefficient, γ_j , defined as

$$f_j^L = \gamma_j x_j f_j^O ; \quad j = 1, 2, \dots, C \quad (4.37)$$

where f_j^O is the fugacity of the j th component at any arbitrarily chosen standard state. Choosing the standard state as the pure component at saturation temperature and pressure, we have

$$f_j^O = \phi_j^{\text{sat}} p_j^{\text{sat}} \approx p_j^{\text{sat}} \quad (4.38)$$

Since ϕ_j^{sat} is close to unity at the saturation temperature of the pure component j . Therefore, from equations 4.16, 4.18 and 4.47, we have at equilibrium,

$$\phi_j^V y_j P = \gamma_j x_j p_j^{\text{sat}} \quad (4.39)$$

$$\text{or } \frac{y_j}{x_j} = K_j \text{ (by definition)} = \frac{\gamma_j p_j^{\text{sat}}}{\phi_j^V P} \quad (4.40)$$

In these methods, the term ϕ_j^V can be calculated, as discussed earlier, using an equation of state or correlations and the term P_j^{sat} is obtained from the Antoine equation (Equation 4.14).

It should be noted that equation 4.40 is structured in a way that the thermodynamic functions have a minimum interdependence on basic variables. Thus ϕ_j^V depends on vapour phase composition, P and T but not on the liquid phase composition. Conversely, γ_j is a function of liquid phase composition only. The saturation pressure P_j^{sat} is only dependent on system temperature, T .

Of the various methods available to compute γ_j , the UNIFAC Group contribution technique is the most recent and widely used. In this method, the activity coefficients in the mixture are related to the interactions between the structural groups. The molecular activity coefficient is divided into two parts: a combinatorial part, which is due to the difference in size and shape of the molecules in the mixture and a residual part which is essentially due to energy interactions among the functional groups in the mixture. For molecule j in the mixture

$$\ln \gamma_j = \ln \gamma_j^C + \ln \gamma_j^R \quad (4.41)$$

Combinatorial Residual

The theoretical aspects and detailed algorithm for calculation of ϕ_j^C and ϕ_j^R have been elaborately dealt by Fredenslund (1977). Table 5 summarizes the final equations.

4.4 Calculation of Enthalpies, $h_{i,j}$ and $H_{i,j}$:

Another thermodynamic property, which is extensively used in distillation calculations, is enthalpy. For liquids, the enthalpy is abbreviated as $h_{i,j}$ and for vapours it is $H_{i,j}$. For convenience sake, we will drop the subscript i in this chapter and denote them by h_j and H_j , respectively.

Curve fit experimental correlation for the enthalpies of all commonly used compounds are available in literature (Owens and Maddox 1968 etc.). Yen and Alexander (1965) have given complicated generalised correlations for most of the compounds for various ranges of conditions. When estimation is required, the enthalpy departure function is calculated from the equations of state. The method of derivation for the departure functions is outlined below. These functions have been listed in Tables 3 and 4. It should be mentioned that they refer to mixture properties and do not have any subscript.

For constant temperature, change in Helmholtz from energy A is given by

$$dA = -PdV \quad (4.42)$$

Integrating this at constant temperature from the reference state volume, V^0 to the system volume V , we have

$$\begin{aligned}
 A-A^{\circ} &= - \int_{V^{\circ}}^V P dV = - \int_{\infty}^V P dV - \int_{V^{\circ}}^{\infty} P dV \\
 &= - \int_{\infty}^V \left(P - \frac{RT}{V} \right) dV - RT \ln \frac{V}{V^{\circ}} \quad (4.43)
 \end{aligned}$$

Other thermodynamic functions can be similarly derived.

The major departure functions of interest are

$$\begin{aligned}
 S-S^{\circ} &= - \frac{\partial}{\partial T} [(A-A^{\circ})]_V \\
 &= \int_{\infty}^V \left[\left(\frac{\partial P}{\partial T} \right)_V - \frac{R}{V} \right] dV + R \ln \frac{V}{V^{\circ}} \quad (4.44)
 \end{aligned}$$

$$\text{and} \quad H-H^{\circ} = (A-A^{\circ}) + T(S-S^{\circ}) + RT(Z-1) \quad (4.45)$$

Thus, when the equation of state to be used is selected, the departure functions can be readily calculated. It must be noted that Eq. 4.45 also applies to liquid enthalpies ($h-h^{\circ}$) when liquid phase mole fractions and compressibility factor are used.

4.5 Calculation of Dew Point (T_D) and Bubble Point (T_B)

The dew point temperature (T_D), at any given pressure, is the temperature at which a vapor of given composition Z_1, Z_2, \dots, Z_C gives the first "drop" of liquid. Similarly, the bubble point temperature (T_B), at any given pressure, is the temperature at which a liquid of composition Z_1, Z_2, \dots, Z_C gives the first 'bubble' of vapour. It has been mentioned earlier that dew

point and bubble point temperatures can be calculated from the following nonlinear equations by using a convergence technique (for example, Regula-Falsi method).

$$f_D = \sum_{j=1}^C \frac{Z_j}{K_j} - 1 = 0 \text{ at dew point with } Z_j = y_j \text{ (given)}$$

$$f_B = \sum_{j=1}^C K_j Z_j - 1 = 0 \text{ at bubble point with } Z_j = x_j \text{ (given)}$$

However, it should be emphasized that in general, the K_j 's depend on both the vapour and liquid compositions and therefore, the computation of dew points and bubble points require an iterative solution over two computational loops - one for the unknown compositions (vapour composition for bubble point estimation and liquid composition for dew point estimation) and the other for the required temperature. Tables 6 and 7 give the flow chart for these computations and problem 3 (Chapter 5) shows the tested example with the results.

TABLE 2

Estimated Basic Physical Properties for Halo-Substituted Silane Compounds

Properties	D	T	M	DH	CL	TH	Ref.
$T_b (^{\circ}K)$	343.15	339.15	331.05	314.15	329.95	305.15	*
M	129.09	149.59	108.59	115.09	170.09	135.59	-
$T_c (^{\circ}K)$	519.779	515.583	499.659	487.853	503.436	475.727	Eq. 4.1
P_c (atm)	33.128	35.012	30.691	37.709	36.456	40.049	Eq. 4.2
α_c	7.079	7.111	7.009	6.866	7.115	6.899	Eq. 4.5
V_c ^{cc} (atm)(gmole)	344.189	322.336	358.901	288.082	302.193	263.895	Eq. 4.4
Z_c	0.2673	0.26675	0.26864	0.2713	0.26667	0.2707	Eq. 4.6
ω	0.2657	0.2722	0.2513	0.2219	0.2729	0.2287	Eq. 4.12

where D = Dimethyldichlorosilane

T = Methyltrichlorosilane

M = Trimethylchlorosilane

DH = Methyldichlorosilane

CL = Tetrachlorosilane

TH = Trichlorosilane

*Values taken from private correspondence (Defence Scientific Information and Documentation Centre, Delhi).

TABLE 3

The Soave - Redlich - Kwong Equation of StateFor Pure Components:

$$Z = \frac{PV}{RT} = \frac{V}{(V-b)} - \frac{\Omega_a}{\Omega_b} \frac{b}{(V+b)} F \quad (4.46)$$

Redlich-Kwong $F = T_r^{-1.5}$

Soave $F = \frac{1}{T_r} \left[1 + (0.48 + 1.574\omega - 0.176\omega^2) (1 - T_r^{0.5}) \right]^2$

$$\Omega_a = 0.4274802327; \quad \Omega_b = 0.086640350$$

For mixtures: (j refers to component number)

$$b_j = \Omega_b \frac{R T_{c_j}}{P_{c_j}}; \quad b = \sum_{j=1}^c Z_j b_j$$

$$F_j = \frac{T_{c_j}}{T} \left[1 + (0.480 + 1.574\omega_j - 0.176\omega_j^2) \left\{ 1 - \left(\frac{T}{T_{c_j}} \right)^{0.5} \right\} \right]^2$$

$$a_j = \frac{\Omega_a}{\Omega_b} \frac{RT}{P} b_j F_j$$

$$F = \frac{\sum_{j=1}^c \sum_{k=1}^c Z_j Z_k (1 - k_{jk}) \left[\frac{T_{c_j} F_j}{P_{c_j}} * \frac{T_{c_k} F_k}{P_{c_k}} \right]^{0.5}}{\sum_{k=1}^c Z_k \frac{T_{c_k}}{P_{c_k}}}$$

$$B_j = \frac{b_j P}{RT};$$

$$B = \sum_{j=1}^c Z_j B_j$$

$$a = \frac{\Omega_a}{\Omega_b} RTFb \quad , \quad A = \frac{\Omega_a}{\Omega_b} FB$$

$$Z^3 - Z^2 + Z(A-B-B^2) - AB = 0 \quad (4.47)$$

$$v = \frac{ZRT}{P}$$

$$\ln \phi_j = \frac{b_j}{b} (Z-1) + \ln \frac{v}{Z(v-b)} + \frac{a}{bRT} \ln \left(\frac{v+b}{v} \right)^*$$

$$\left[\frac{b_j}{b} - 2 \sum_{K=1}^c (1-K_{jk}) \frac{(a_j a_k)^{0.5}}{a} Z_K \right] \quad (4.48)$$

Z_j = liquid phase mole fraction to get ϕ_j^L
 = vapour phase mole fraction to get ϕ_j^V

Expressions for the Departure Functions:

Molar Free Energy A:

$$A-A^\circ = RT \left[-\ln \left(\frac{v-b}{v} \right) - \frac{\beta}{b} \ln \left(\frac{v+b}{v} \right) - \ln \left(\frac{v}{v^\circ} \right) \right] \quad (4.49)$$

where

$$\beta = \sum_{j=1}^c \sum_{K=1}^c Z_j Z_K (1-K_{jk}) \left[\left(\frac{\Omega_a}{\Omega_b} \right)^* \left(\frac{\Omega_a}{\Omega_b} \right)_{b_j b_k F_j F_k} \right]^{0.5}$$

$$v^\circ = (RT/P^\circ)$$

Molar Entropy S:

$$S-S^\circ = R \left(\ln \left(\frac{v-b}{v} \right) - \frac{\gamma}{b} * \ln \left(\frac{v+b}{v} \right) + \ln \frac{v}{v^\circ} \right) \quad (4.50)$$

where $\gamma = -\frac{1}{2} \sum_{j=1}^c \sum_{k=1}^c Z_j Z_k (1-K_{jk}) \left[\frac{\Omega_a}{\Omega_b} b_j b_k F_j F_k \right]^{0.5} *$

$$\left[\frac{f(\omega_k)}{F_k^{0.5}} + \frac{f(\omega_j)}{F_j^{0.5}} \right]$$

$$f(\omega_j) = 0.480 + 1.574 \omega_j - 0.176 \omega_j^2$$

$$S^\circ = \int_{T^\circ}^T \frac{C_p}{T} dT$$

Molar Enthalpy H:

$$H-H^\circ = (A-A^\circ) + T(S-S^\circ) + RT(Z-1)$$

$$H^\circ = \int_{T^\circ}^T C_p dT \quad (\text{i.e. Zero at standard condition chosen, } T^\circ)$$

$$P^\circ = 1 \text{ atmosphere.}$$

TABLE 4The Peng - Robinson Equation of StateFor Pure Components:

$$P = \frac{RT}{(v-b)} - \frac{a}{v(v+b) + b(v-b)} \quad (4.51)$$

where a and b are constants

For mixtures: (j denotes component numbers)

$$b_j = 0.7780 \frac{RT_{c_j}}{P_{c_j}}$$

$$b = \sum_{j=1}^c Z_j b_j$$

$$a_{c,j} = 0.47724 \frac{R^2 T_{c_j}^2}{P_{c_j}}$$

$$\alpha_j^{0.5} = 1 + (0.37464 + 1.54226 \omega_j - 0.26992 \omega_j^2)(1 - T_r^{0.5})$$

$$a_j = a_{c,j} \alpha_j$$

$$a = \sum_{j=1}^c \sum_{k=1}^c Z_j Z_k (1 - K_{jk}) a_j^{0.5} a_k^{0.5}$$

where K_{jk} 's are the binary interaction parameters (They are taken to be unity where sufficient data are not available).

$$B = \frac{bP}{RT} \quad ; \quad A = \frac{aP}{R^2 T^2}$$

$$Z^3 + (B-1)Z^2 + (A-3B^2-2B)Z - (AB-B^2-B^3) = 0 \quad (4.52)$$

Table 4 (contd)

$$\begin{aligned}
 v &= ZRT/P \\
 a_{kj}^* &= (1 - K_{kj}) a_k^{0.5} a_j^{0.5} \\
 \ln \phi_j &= \frac{b_j}{b} (Z-1) - \ln(Z-b) \\
 &\quad - \frac{A}{2.828B} \left[\frac{2 \sum_{k=1}^c Z_k a_{kj}^*}{a} - \frac{b_j}{b} \right] \ln \left(\frac{Z+2.414B}{Z-0.414B} \right) \quad (4.53)
 \end{aligned}$$

where Z_j = liquid composition to get ϕ_j^L and
 = vapour composition to get ϕ_j^V

Expressions for Departure Functions for Mixtures:

$$\begin{aligned}
 f(\omega_j) &= 0.37464 + 1.54226 \omega_j - 0.26992 \omega_j^2 \\
 \frac{da}{dT} &= -\frac{1}{2T} \sum_{j=1}^c \sum_{k=1}^c Z_j Z_k (0.45724^{0.5} R) \left[\frac{T_r^{0.5} T_{c_k} f(\omega_k)}{P_{c_k}^{0.5}} a_j^{0.5} \right. \\
 &\quad \left. + \frac{T_r^{0.5} T_{c_j} f(\omega_j)}{P_{c_j}^{0.5}} a_k^{0.5} \right] \\
 S-S^0 &= \left(\frac{da}{dT} \right) \ln \left(\frac{Z+2.414B}{Z-0.414B} \right) - R \ln \left(\frac{v^0}{v-b} \right) \quad (4.54)
 \end{aligned}$$

where, S denotes entropy of the mixture

$$H-H^0 = RT(Z-1) + \frac{(T \frac{da}{dT} - a)}{(2\sqrt{2}b)} \ln \frac{(Z+2.414B)}{(Z-0.414B)} \quad (4.55)$$

where, H denotes enthalpy of the mixture.

$$\text{and } A-A^{\circ} = (H-H^{\circ}) - T(S-S^{\circ}) - RT(Z-1)$$

$$\text{with } S^{\circ} = \int_{298^{\circ}\text{K}}^T \frac{C_p}{T} dT$$

$$H^{\circ} = \int_{298^{\circ}\text{K}}^T C_p dT$$

$$v^{\circ} = RT/P^{\circ} ; \quad P^{\circ} = 1 \text{ atmosphere}$$

TABLE 5

Unifac Group Contribution Equations

$$\ln \gamma_j = \ln \gamma_j^C + \ln \gamma_j^R \quad (4.56)$$

Combinatorial Residual

$$\ln \gamma_j^C = \ln \frac{\phi_j}{x_j} + \frac{Z}{2} q_j \ln \frac{\theta_j}{\phi_j} + l_j - \frac{\phi_j}{x_j} \sum_{j=1}^c x_j l_j \quad (4.57)$$

where

$$l_j = \frac{Z}{2}(r_j - q_j) - (r_j - 1); \quad Z=10$$

$$\theta_j = \frac{q_j x_j}{\sum_{j=1}^c q_j x_j}; \quad \phi_j = \frac{r_j x_j}{\sum_{j=1}^c r_j x_j}$$

Molecular surface
area fraction

Molecular volume
fraction

The vander Waals volume $r_j = \sum_{k=1}^c \sum_k^{(i)} R_k$

The vander Waals surface area $q_i = \sum_{k=1}^N \sum_k^{(i)} Q_k$

where N = number of groups in molecule i

R_k and Q_k are group volume and area parameters tabulated in Fredenslund (1977)

$$\ln \gamma_j^R = \sum_k \sum_k^{(i)} \ln \Gamma_k - \ln \Gamma_k^{(i)} \quad (4.58)$$

all groups

$K = 1, 2, \dots, N$ (where N = number of different groups in the mixture)

Γ_k = residual activity coefficient of group k in a solution

$\Gamma_k^{(i)}$ = residual activity coefficient of group k in a reference solution containing only molecules of type i

$$\ln \Gamma_k = Q_k \left[1 - \ln \left(\sum_m \phi_m \psi_{mk} \right) - \sum_m \left(\phi_m \psi_{km} / \sum_n \phi_n \psi_{nm} \right) \right] \quad (4.59)$$

m and n = 1, 2, ..., N (all groups)

Eqn. 4.54 also holds for $\Gamma_k^{(i)}$

$$\phi_m = \frac{Q_m x_m}{\sum_{n=1}^N Q_n x_n} ;$$

$$x_m = \frac{\sum_{p=1}^c \sum_{n=1}^N \psi_n^{(p)} x_p}{\sum_{p=1}^c \sum_{n=1}^N \psi_n^{(p)} x_p} \quad (4.60)$$

Group surface area fraction

Group fraction

$$\psi_{nm} = \exp (-a_{nm}/T)$$

where a_{nm} is the binary group interaction parameter given in Fredenslund (1977). It should be noted that there are two such parameters for each pair of groups and $a_{nm} \neq a_{mn}$

TABLE 6

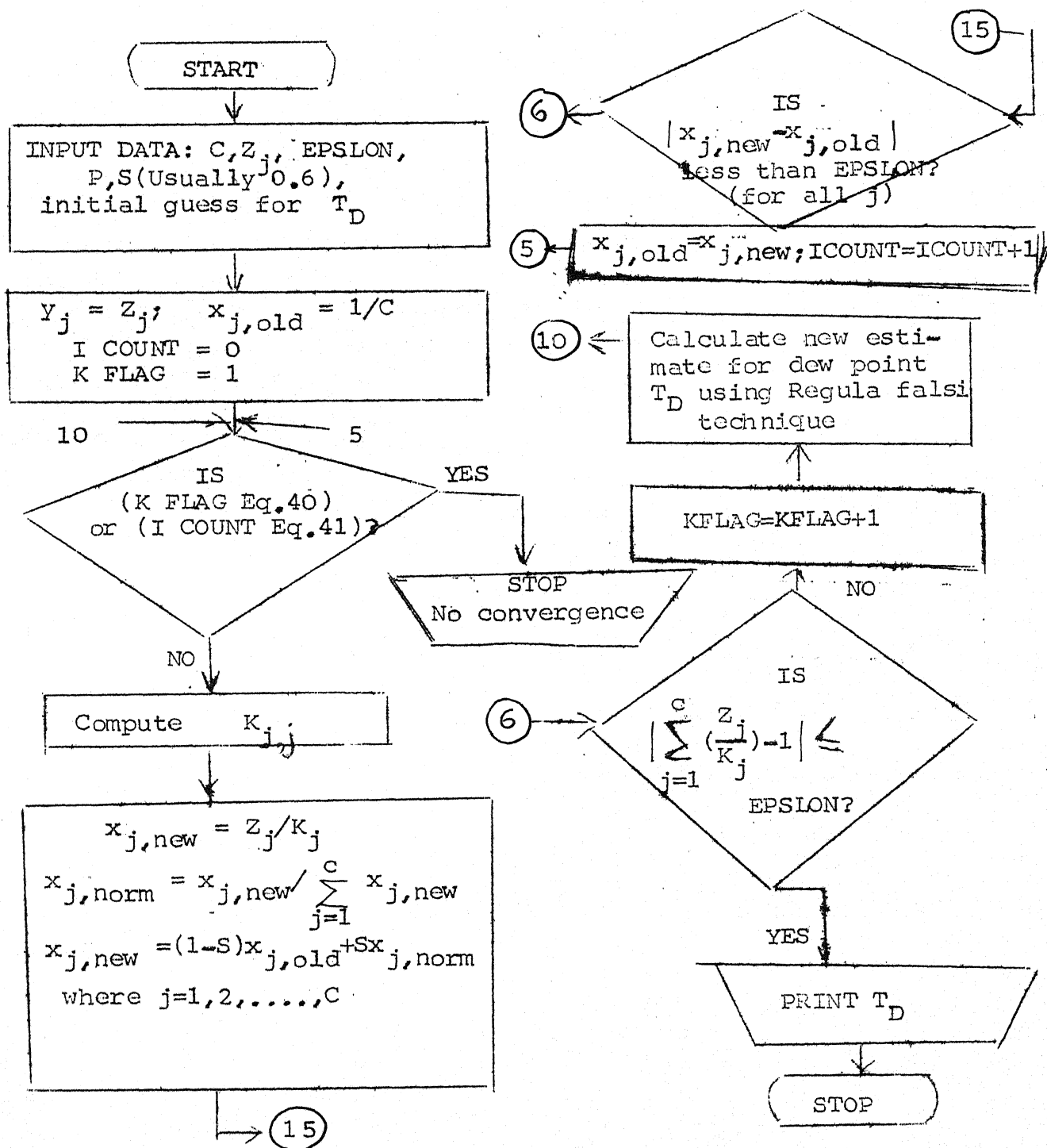
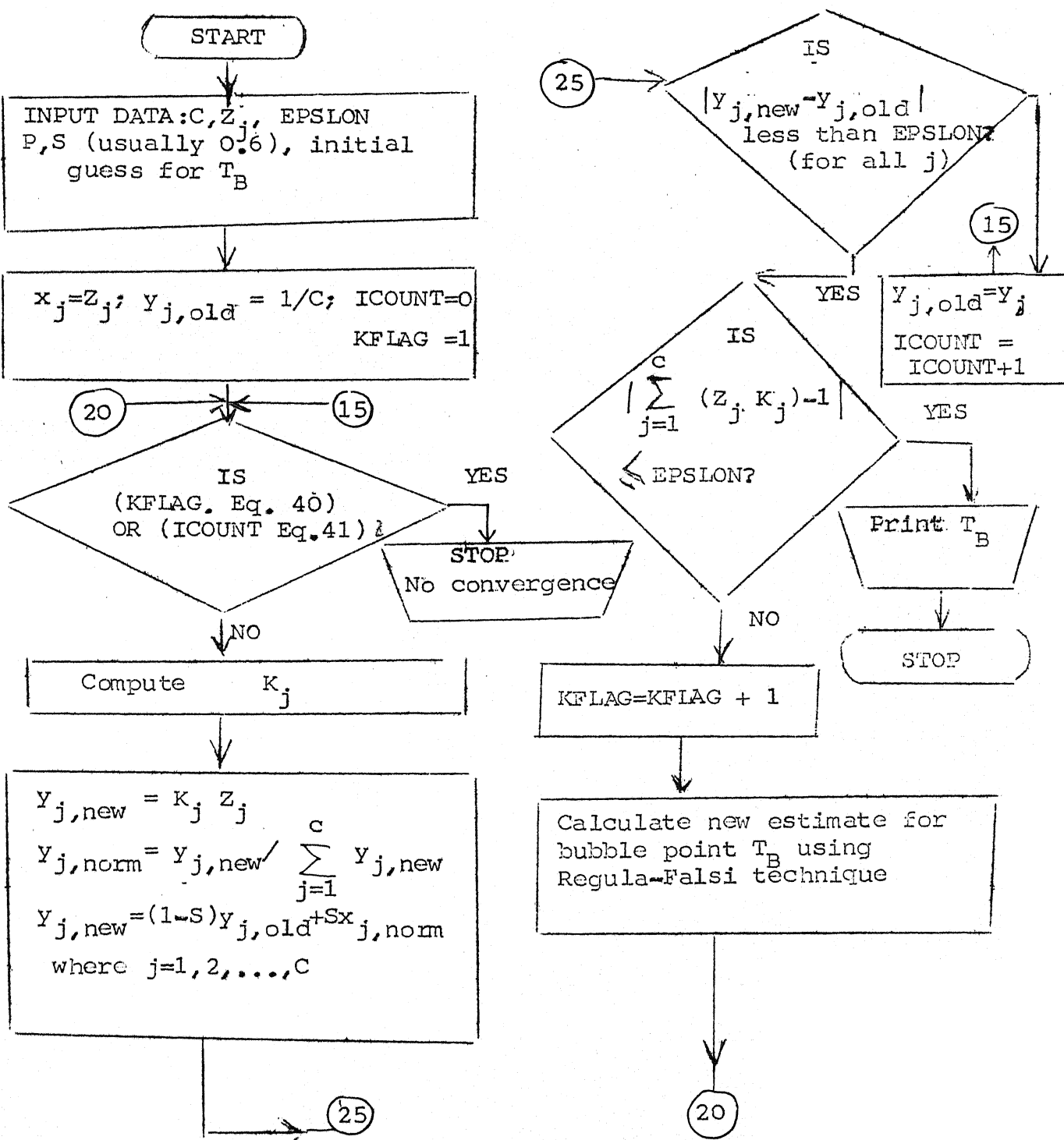
Flowchart for Dew Point Temperature (T_D) Estimation

TABLE 7

Flow Chart for Bubble Point Temperature (T_B)Estimation

CHAPTER 5

RESULTS AND DISCUSSION

Several test problems have been solved using the algorithms developed during the present work. Through these examples an attempt has been made to check the programs for VLE data prediction, basic thermodynamic property estimation, dew point and bubble point estimations, distillation column simulation and design having numerical schemes like Newton-Raphson, Regula-Falsi, Thomas Algorithm, Block Thomas Algorithm, Gaussian Elimination, Golden Section Search etc. Some of these problem categories will be presented in this chapter.

They are as follows:

Problem 1: Estimation of K-values for nearly ideal and nonideal systems

Problem 2: Binary VLE data prediction for Halogen-substituted silane systems

Problem 3: Dew point and Bubble point calculations for ideal and nonideal systems

Problem 4: Distillation Column Simulation using ideal and nonideal thermodynamics

Problem 5: Distillation column design using the new simultaneous semi-tray-by-tray formulation.

Comparison has been made of computed results with published data wherever possible which validates the developed programs.

5.1 VLE Data Prediction (Problem 1 and 2):

Problem 1 demonstrates the estimation of K-values for common systems for which all the critical values are available. Both Soave-Redlich-Kwong (SRK) and Peng-Robinson equations of state have been used for prediction of the K-values. Available experimental values which are used for comparison are quite reliable. The agreement between the two is generally good. The following observations can however be made:

- (i) the accuracy for prediction of VLE data by equations of state decreases at conditions near and above the critical point (e.g. methane in problem 1.1).
- (ii) the accuracy of prediction also decreases for compounds having associations (e.g. methanol in problem 1.3). This observation has also been made by many other authors.
(Fredenslund, 1977, Srinivasan, 1982 etc.)

Problem 2 presents the prediction of K-values for systems (Halosubstituted silanes) for which critical properties are not available and have been estimated by the procedure outlined earlier. The input data are molecular weights and boiling points of the various compounds. Only meagre experimental data is available on these systems and that too with

limited reliability. The agreement between predicted results and experimentally observed values is generally good except in case of silicon tetrachloride and trimethylchlorosilane (at certain conditions) for which the equation of state may be less applicable or the observed values may be less certain.

Problem 11: Estimation of Partition Coefficients (K-values) for Nearly Ideal System

System: A mixture of Methane (1)

Ethane (2)

Propane (3)

Temperature: 283.15 K ; Pressure: 13.61 atmosphere

$$x_1 = 0.029 \quad ; \quad y_1 = 0.290$$

$$x_2 = 0.166 \quad ; \quad y_2 = 0.292$$

$$x_3 = 0.805 \quad ; \quad y_3 = 0.418$$

Physical Properties

Components	M	T_c (°K)	P_c (atm)	V_c (cm ³ /g.mole)	Z_c	ω
1	16.043	190.6	45.4	99.0	0.288	0.008
2	30.07	305.4	48.2	148.0	0.285	0.098
3	44.097	369.8	41.9	203.0	0.281	0.152

Comparison of K_j values:

Component	Predicted K_j values using		Experimental values*
	SRK Eq.	Peng-Robinson Eq.	
1	9.2	9.0	10.0
2	1.82	1.8	1.76
3	0.545	0.543	0.52

* Taken from Smith and Vanness (1975)

CPU time: 0.07 sec

Elapsed time: 0.76 sec.

Problem 1.2: Estimation of Partition Coefficients for Nonideal System:

System: A mixture of Benzene (1)

Cyclohexane (2)

n-hexane (3)

Temperature: 342.7 K ; Pressure: 1 atmosphere

$x_1 = 0.279$; $y_1 = 0.250$

$x_2 = 0.037$; $y_2 = 0.025$

$x_3 = 0.684$; $y_3 = 0.725$

Physical Properties:

Component	M	T_c (°K)	P_c (atm)	V_c (cm ³ /g.mole)	Z_c	ω
1	78.114	562.1	48.3	259.0	0.271	0.212
2	84.162	553.4	40.2	308.0	0.273	0.213
3	86.178	507.4	29.3	370.0	0.260	0.296

Comparison of K_j Values:

Component	Predicted K_j values using		Experimental values*
	SRK Eq.	Peng. Robinson Eq	
1	0.8445	0.8605	0.8960
2	0.7442	0.7551	0.6756
3	1.0375	1.0397	1.0599

* Taken from Srinivasan (1982)

Problem 1.3: Estimation of Partition Coefficients for a Highly Nonideal System:

System: A mixture of Methanol (1)

Carbontetrachloride (2)

Benzene (3)

Temperature: 328.15°K ; Pressure = 0.9296 atmospheres

$x_1 = 0.194$; $y_1 = 0.507$

$x_2 = 0.592$; $y_2 = 0.373$

$x_3 = 0.213$; $y_3 = 0.12$

Physical Properties

Component	M	T_c ($^{\circ}\text{K}$)	P_c (atm)	V_c ($\text{cm}^3/\text{g.mole}$)	Z_c	ω
1	32.042	512.6	79.9	118.0	0.224	0.559
2	153.823	556.4	45.0	276.0	0.272	0.194
3	78.114	562.1	48.3	259.0	0.271	0.212

Comparison of K_j values:

Component	Predicted K_j values using		Experimental Values*
	SRK Eq.	Peng-Robinson Eq.	
1	0.97	1.015	2.613
2	0.5462	0.5601	0.63
3	0.4697	0.483	0.5634

* Taken from Fredenslund et al. (1977) and Srinivasan (1982)

Problem 2.1: Binary VLE Data Prediction for Halosubstituted Silane Systems

(a) System: A mixture of Trimethylchlorosilane (1)
and Methyltrichlorosilane (2)

Pressure: 1 atmosphere

Physical properties: taken from Table 2

Comparison of K_j values:

Conditions		Component	Predicted K_j values using		Experimental values*
Temperature (°K)	Mole fraction		SRK Eq.	Peng-Robinson Eq.	
338.05	$x_1=0.1$	1	1.2446	1.2455	1.19
	$y_1=0.119$	2	0.9443	0.9499	0.9789
337.25	$x_1=0.25$	1	1.2046	1.2059	1.18
	$y_1=0.295$	2	0.9229	0.9288	0.94
335.35	$x_1=0.50$	1	1.1266	1.1288	1.136
	$y_1=0.568$	2	0.8754	0.8817	0.864
333.05	$x_1=0.75$	1	1.0454	1.0486	1.0747
	$y_1=0.806$	2	0.8221	0.8290	0.776
331.57	$x_1=0.9$	1	0.9980	1.0017	1.0283
	$y_1=0.926$	2	0.7900	0.7972	0.74

* Taken from Zel'vensky (1971)

(b) System: A mixture of Trimethylchlorosilane (1)

and Dimethyldichlorosilane (2)

Pressure: 1 atmosphere

Physical properties: taken from Table 2

Comparison of K_j values

Conditions		Components	Predicted K_j values using j		Experimental Values*
Temperature (°K)	Mole fraction		SRK Eq	Peng- Robinson Eq	
342.15	$x_1=0.1$	1	1.072	1.0764	1.35
	$y_1=0.135$	2	0.95	0.9549	0.9611
340.35	$x_1=0.25$	1	1.0145	1.02	1.248
	$y_1=0.312$	2	0.8993	0.905	0.9173
337.17	$x_1=0.5$	1	0.9189	0.925	1.2
	$y_1=0.6$	2	0.815	0.8213	0.8
333.93	$x_1=0.75$	1	0.829	0.8356	1.1026
	$y_1=0.827$	2	0.7357	0.74	0.692
331.95	$x_1=0.9$	1	0.7775	0.7846	1.0411
	$y_1=0.937$	2	0.69	0.697	0.63

* Taken from Zel'vensky (1971)

(c) System: A mixture of Methyltrichlorosilane (1)

and Tetrachlorosilane (2)

Pressure: 1 atmosphere

Physical properties: taken from Table 2

Comparison of K_j values

Conditions		Components	Predicted K_j values using		Experimental values*
Temperature (°K)	Mole fractions		SRK Eq.	Peng-Robinson Eq.	
337.05	$x_1=0.9$	1	0.9149	0.9209	0.95
	$y_1=0.855$	2	1.2151	1.2184	1.45
335.09	$x_1=0.75$	1	0.8601	0.8665	0.9013
	$y_1=0.676$	2	1.1448	1.1489	1.296
332.62	$x_1=0.5$	1	0.7946	0.8015	0.866
	$y_1=0.433$	2	1.0606	1.0655	1.134
331.30	$x_1=0.25$	1	0.7612	0.7684	0.844
	$y_1=0.211$	2	1.0175	1.0229	1.052
330.40	$x_1=0.1$	1	0.739	0.7464	0.83
	$y_1=0.083$	2	0.9889	0.9946	1.0189

* Taken from Zel'vensky (1971)

5.2 Dew Point and Bubble Point Calculations (Problem 3):

Calculations have been made using ideal as well as nonideal thermodynamics. In problem 3.1(a) and 3.2(a) ideal thermodynamics is used to estimate the dew point and bubble point temperatures of vapour and liquid which are themselves in the thermodynamic equilibrium with each other. As shown below, the results are in close agreement with published values (Amu. . . 1958). In problems 3.1(a) and 3.2(b) the system is same as that in part (a) except that nonideal thermodynamics is used to calculate K-values. This, however, changes the mole fractions of various components in the two phases which are in thermodynamic equilibrium. The comparison of results with part (a) shows how these temperatures vary when nonideality is introduced. Problems 3.1(c) and 3.2(c) use the same vapour and liquid compositions as in 3.1(a) and 3.2(a) and proceed to calculate the two temperatures using nonideal thermodynamics. These compositions are different from these in part (b) where the compositions of vapour and liquid themselves were calculated using nonideal thermodynamics.

Problem 3.1(a)

Feed composition:

$n\text{-C}_3(1)$:	130.4210 lb.moles/hr
$n\text{-C}_4(2)$:	12.7402 lb.moles/hr
$n\text{-C}_5(3)$:	0.7588×10^{-1} lb.moles/hr

Thermodynamics used : Ideal

Initial guesses supplied: 120°F and 170°F

Calculated dew point temperature: 144.044°F

Literature values : 140°F

CPU time : 0.07 seconds

Elapsed time : 3.10 seconds

Tolerance limit : 0.0001

Problem 3.1(b)

Feed composition:

n-C₃ (1) : 121.739 lb.moles/hr

n-C₄ (2) : 14.1454 "

n-C₅ (5) : 0.20218 "

Thermodynamics used : Nonideal

Initial guesses supplied: 120°F and 170°F

The liquid mole fractions
estimated

: (1) $x_1 = 0.773985$

(2) $x_2 = 0.2195954$

(3) $x_3 = 6.41947 \times 10^{-3}$

Calculated dew point temperature: 140.64°F

Number of iterations : 4

CPU time : 1.47 seconds

Elapsed time : 27.56 seconds

Tolerance limit : 0.0001

Problem 3.1(c)

Feed composition : Same as in 3.1(a)

Thermodynamics used : Nonideal

Initial guesses : same as in 3.1(a)

No. of iterations : 4

Calculated dew point

temperature : 137.87°F
CPU time : 1.46 seconds
Elapsed time : 29.82 seconds
Tolerance limit : 0.0001

Problem 3.2(a)

Feed composition:

n-C₃(1) : 63.7496 lb.moles/hr
n-C₄(2) : 40.077 "
n-C₅(3) : 2.5466 "

Thermodynamics used : Ideal

Initial guesses supplied : 150°F and 180°F

Calculated bubble point
temperature : 163.7533°F

Literature value : 160.53°F

CPU time : 0.05 seconds

Elapsed time : 2.98 seconds

Tolerance limit : 0.0001

Problem 3.2(b)

Feed Composition:

n-C₃(1) : 46.7383 lb.moles/hr
n-C₄(2) : 45.6364 lb.moles/hr
n-C₅(3) : 4.9448 lb.moles/hr

Thermodynamics used : Nonideal

Initial guesses supplied : 150°F and 190°F

The vapour mole fractions
estimated : $y_1 = 0.6871599$

$y_2 = 0.2964676$

$y_3 = 0.0163724$

Calculated bubble point
temperature : 170.6868°F

Number of iterations	: 3
CPU Time	: 1.02 seconds
Elapsed time	: 20.56 seconds
Tolerance limit	: 0.0001
Problem 3.2(c)	
Feed Composition	: same as 3.2(a)
Thermodynamics used	: Nonideal
Initial guesses	: 150°F and 180°F
Number of iterations	: 2
Calculated bubble point temperature	: 157.3949°F
CPU time	: 0.62 seconds
Elapsed time	: 2.52 seconds
Tolerance limit	: 0.0001

5.3 Simulation of Distillation Columns (Problem 4):

The multicomponent, multistage distillation column calculation package has been used in simulation (or rating) mode without any multitray models to check its performance. Problem 4.1 deals with a 3 component feed to a 16 stage column (including reboiler and condenser). Both vapours and liquid are assumed to behave as ideal fluids. The calculated component flowrates of distillate and bottoms match very well with the published results (Amundson, 1958).

Problem 4.2 uses the same feed as in Problem 4.1 and the same column configuration, but both vapours and liquid are treated as nonideal and the single equation of state approach

is used to account for the nonideality. The component flow rates obtained do not vary significantly from those in Problem 4.1 primarily because the feed mixture is near ideal. However, the temperature profiles in the two problems vary to a larger extent than the compositions (see Appendix 6).

Problem 4.3 has been used with same feed composition as before but with a different column configuration (13 stages). Ideal thermodynamics has been used. As expected, the separation achieved has decreased in this case as compared to problem 4.1.

Feed rate	: 100.0 lb.moles/hr with
Feed composition (component flow rate)	: (1) $n\text{-C}_3$: 23.0 lb.moles/hr (2) $n\text{-C}_4$: 37.0 (3) $n\text{-C}_5$: 40.0
Feed condition	: Saturated liquid
Feed temperature	: 225.0°F
Reflux ratio	: 5
Distillate rate	: 22.6 lb.moles/hr
Condenser temperature	: 120.0°F
Condenser type	: Total condenser

Problem 4.1 Number of stage : 16

Feed location : 8th tray from the top

The following top and bottom flow rates were obtained using ideal thermodynamics.

Component	Distillate, $v_{1,j}$	Bottoms, $l_{16,j}$
n-C ₃	21.5360 lb.moles/hr	1.4637 lb.moles/hr
n-C ₄	1.0614 ,,	35.9385 ,,
n-C ₅	0.0022 ,,	39.9978 ,,
nTotal	22.5996 lb.moles/hr	77.4000 lb.moles/hr

Optimal values of β (in modified Newton-Raphson method) were found to be between .98 and .9999 for different iterations.

Tolerance limit : 0.0001
 No. of iteration : 3
 CPU time : 4.28 seconds
 Elapsed time : 30.58 seconds

The vapour and liquid flow and temperature profiles obtained in the column have been provided in Appendix G.

Problem 4.2: Number of stages : 16

Feed location : 8th tray from the top

The following top and bottom flow rates were obtained using nonideal thermodynamics.

Component	Distillate, $v_{1,j}$	Bottoms $l_{16,j}$
n-C ₃	21.47325 lb.moles/hr	1.5267 lb.moles/hr
n-C ₄	1.11868 ,,	35.8813 ,,
n-C ₅	0.00806 ,,	39.9919 ,,
Total	22.59999 lb.moles/hr	77.3999 lb.moles/hr

Optimal values of β (in modified Newton-Raphson method) were found to be between 0.94 and 0.999 for different iterations.

Tolerance limit	: 0.0001
Number of iterations	: 5
CPU time	: 25.54 seconds
Elapsed time	: 1 minute 29.52 seconds

The output listing for the variable profiles obtained in the column has been given in Appendix G.

Problem 4.3: Number of stages: 13

Feed location : 7th tray from the top

The following top and bottom flow rates were obtained using ideal thermodynamics.

<u>Component</u>	<u>Distillate, $v_{1,j}$</u>	<u>Bottoms $l_{13,j}$</u>
n-C ₃	20.8177 lb.moles/hr	2.1888 lb.moles/hr
n-C ₄	1.7796 lb.moles/hr	35.2203 lb.moles/hr
n-C ₅	0.0092 lb.moles/hr	39.9908 lb.moles/hr
<u>Total</u>	<u>22.6065 lb.moles/hr</u>	<u>77.3999 lb.moles/hr</u>

Optimal values of β (in modified Newton-Raphson method) were found to be between 0.98 and 0.9999 for different iterations.

Tolerance limit	: 0.0001
No. of iterations	: 3
CPU time	: 3.96 seconds
Elapsed time	: 35.8 seconds

The output listing for the variable profiles obtained in the column has been given in Appendix G.

5.4 Design of Distillation Columns (Problem 5):

The earlier problems 4.1 and 4.3 have been solved in the design mode i.e., where the number of stages and feed locations have been treated as unknowns. The vapor flow rate of component 1 on unit number . and vapour flow rate of component 3 on unit number were used as design specification. In problem 5.1, the total number of stages was found to be 16.04 to be compared with 16 trays (problem 4.1) and feed location at 8.2 to be compared with 8. The Edmister approximation seems to be quite satisfactory for this problem as evidenced by a very small variation obtained, as compared to actual one.

In problem 5.2 the total stages obtained is 13.02 to be compared with actual number of 13 plates and feed location at 7th plate exactly same as actual.

Total feed rate : 100.0 lb.moles/hr with
component feed rate (1) n-C₃ : 23.0 lb.moles/hr
(2) n-C₄ : 37.0 lb.moles/hr
(3) n-C₅ : 40.0 lb.moles/hr

Feed condition: Saturated liquid

Feed temperature : 225.0°F

Thermodynamics used : Ideal

Specifications:

Reflux ratio : 5

Distillate rate : 22.6 lb.moles/hr

Condenser temperature: 120°F

Problem 5.1:

Vapour flow rate for component

1 on unit 3 : 130.42100 lb.moles/hr

Vapour flow rate for component

3 on unit 9 : 17.374790 lb.moles/hr

The following top and bottom flow rates were obtained.

<u>Component</u>	<u>Distillate, $v_{1,j}$</u>	<u>Bottoms, $l_{11,j}$</u>
$n-C_3$	21.5383 lb.moles/hr	1.4617 lb.moles/hr
$n-C_4$	1.0590 lb.moles/hr	35.9409 lb.moles/hr
$n-C_5$	0.0027 lb.moles/hr	39.9973 lb.moles/hr
<u>Total</u>	<u>22.6000 lb.moles/hr</u>	<u>77.39999 lb.moles/hr</u>

Total number of trays : 16.04

Feed location at 8.2 th tray from the top, optimal β was found to be between 0.96 and 0.9999

Tolerance limit : 0.0001

Number of iterations : 5

CPU time : 20.62 seconds

Elapsed time : 58.5 seconds

Problem 5.2:

Vapour flowrate for component 1 on

unit 3 : 123.03 lb.moles/hr

Vapour flow rate for component 3

on unit 9 : 18.8633 lb.moles/hr

The following top and bottom flow rates were obtained:

<u>Component</u>	<u>Distillate, $v_{1,j}$</u>	<u>Bottoms, $l_{11,j}$</u>
n-C ₃	20.8114 lb.moles/hr	2.1386 lb.moles/hr
n-C ₄	1.7794 lb.moles/hr	35.2206 lb.moles/hr
n-C ₅	0.0092 lb.moles/hr	39.9907 lb.moles/hr
<u>Total</u>	<u>22.6000 lb.moles/hr</u>	<u>77.3999 lb.moles/hr</u>

Total number of trays : 13.02

Feed location at 7th tray from the top. Optimal β was found to be between 0.96 and 0.9999

Tolerance limit : 0.0001

Number of iterations : 5

CPU time : 20.62 seconds

Elapsed time : 58.5 seconds

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

In the present work, multistage, multicomponent distillation column design and simulation procedure has been developed. The existing literature lacked a generalized doubly-acting algorithm which could be applied to various column configurations and take into account nonidealities with respect to thermodynamics and plate efficiencies. The proposed formulation has been found to overcome most of the shortcomings. A combination of tray-by-tray and shortcut methods used in the present study enables the user to design a column of any desired configuration without repeated application of simulation procedure as previously required. The shortcut method applied to the multitray models has been made as rigorous as practical and most of the simplifying assumptions have been removed. The numerical aspects of the formulation have been selected and implemented in away that computation time and memory requirements are minimum. The computer package has been written in FORTRAN language and implemented on DEC-1090 system. Some problems from literature have been tested and good convergence characteristics have been obtained. The CPU time and memory requirements are almost independent of the number of trays.

It was found that about 60 percent of the CPU time required for the computations was taken up by the numerical Jacobian calculations. Instead of evaluating the Jacobian numerically in every iteration, an alternate procedure (Kaufmann, 1983) of updating the Jacobian from the previous iterations can be experimented upon. However, it should be recognized that using the previous iteration Jacobians will increase memory requirement since in that case it will have to be stored in the memory which in the present method is not necessary. Extensive variable packing schemes could be used to reduce memory requirements which have not been attempted in the existing program for the sake of clarity of understanding.

The effective stripping factor of Edmister may be treated as a function of the number of stages inside the multitray unit, M_1 when its size becomes too large (Ohmura, 1978) or the size of the multitray unit may be shrunk with a corresponding increase in the single tray unit. The thermodynamicss for associated compounds and highly nonideal systems and property estimation near critical conditions should be further studied and improved upon. Calculations for a variety of operating industrial distillation problems are required to be tested and checked against some reputed simulator packages (e.g., Chemshare) regarding performance, memory and CPU time. This should also build confidence in the present package with respect to its design capability.

REFERENCES

- Adler, S.B., Spencer, C.F., Kuo, C.M. and Ozkardesh, H.; American Chemical Society Symposium Series, 60, Washington (1977).
- Amundson, N.R. and Pontinen, A.J.: I&EC, 50, 730 (1958).
- Antoine, C.: C.R., 107, 681, 836 (1888) in Reid (1977).
- Beveridge, G.S.G. and Schechter, R.S.: "Optimization Theory and Practice", McGraw Hill Co. (1970).
- Boynnton, G.W.: Hydrocarbon Process. 49(1), 153 (1970).
- Chakraborty, A.: M.Tech. Dissertation, I.I.T. Kanpur (1983).
- Das, A.K.: M.Tech. Dissertation, I.I.T. Kanpur (1985).
- Edmister, W.C.: I&EC, 35, 837 (1943).
- Edmister, W.C.: Petrol. Engg., 19(8), 128 (1948).
- Edmister, W.C.: Petrol. Engg., 19(9), 47 (1948).
- Erbar, J.H. and Maddox, R.N.: Petrol Refin., 40(5), 183 (1961).
- Fenske, M.R.: I&EC, 24, 482 (1932).
- Fredenslund, A., Gmehling, J., Michelson, M.L., Rasmussen, P. Prausnitz, J.M.: I&EC, Proc. Des. & Dev. 16, 450 (1977).
- Fredenslund, A., Gmehling, J. and Rasmussen, P.: "Vapor-Liquid Equilibria Using UNIFAC" ELSEVIER Scientific Publishing Co. (1977)
- Gallun, S.E. and Holland, C.D.: Comp. & Chem. Eng., 4(2), 93 (1980).
- Gilliland, E.R.: I&EC, 32, 1220 (1940).
- Harbert, W.D.: I & EC, 37, 1162 (1945).
- Hofeling, B.S. and Seader, J.D.: AIChE J., 24(6), 1131 (1978).

Holland, C.D.: "Fundamentals and Modeling of Separation Processes"
Prentice Hall, Inc. (1975).

Holland, C.D.: "Fundamentals of Multicomponent Distillation"
McGraw Hill, Inc. (1981).

Kaufmann, S. and Montalvo, A.: Comp. & Chem. Eng., 7(2), 93 (1983).

Ketchum, R.G.: Chem. Eng. Sci., 34, 387 (1979).

King, C.J.: "Separation Processes", McGraw Hill, Inc. (1980).

Kudchadkar, A.P., Alani, G.H. and Zwolinski, B.J.: Chem. Rev.,
68, 659 (1968).

Lewis, W.K. and Matheson, G.L., I&EC, 24, 444 (1932).

Lydersen, A.L.: Eng. Exp. Stn. Rep. 3, Univ. Wisconsin Coll.
Eng., April 1955 in Reid (1977).

Mathews, J.F.: Chem. Rev., 72, 71 (1972).

Naphtali, L.M. and Sandholm, D.P.: AIChE J., 17, 148 (1971).

Ohmura, S. and Kasahara, S.: J. Chem. Eng. Japan, 11(3), 185 (1978).

Ohmura, S. and Nishizawa, K. J. Japan Petrol. Inst., 21, 94 (1978).

Ohmura, S.: Japan Petrol. Inst., 21(5), 297 (1978).

Owens, W.R. & Maddox, R.N.: I&EC, 60 (12), 15 (1968).

Perry, J.H.: Chemical Engineers Handbook, 4th ed., McGraw Hill,
1963.

Pitzer, K.S.: J. Am. Chem. Soc., 77, 3427 (1955).

Reid, R.C., Prausnitz, J.M. and Sherwood, T.K.: "The Properties
of Gases and Liquids", McGraw Hill Co. (1977).

Ricker, N.L. and Grens, E.A.: AIChE J., 20, 238 (1974).

Riedel, L.: Chem. Ing. Tech., 26, 83 (1954).

Smith, B.D.: "Design of Equilibrium Stage Processes", McGraw Hill, Inc. (1963)

Soave, G.S.: Chem. Eng. Sci., 27, 1195 (1972).

Srinivasan, R.: M.Tech. Dissertation, I.I.T. Kanpur (1982).

Stull, D.R.: I&EC, 39, 517 (1947).

Tewarson: "Sparse Matrices", Academic Press Inc. (1973).

Tomich, J.F.: AIChE J., 16, 2 (1970).

Underwood, A.V.J.: Trans. Inst. Chem. Eng. 10, 112 (1932).

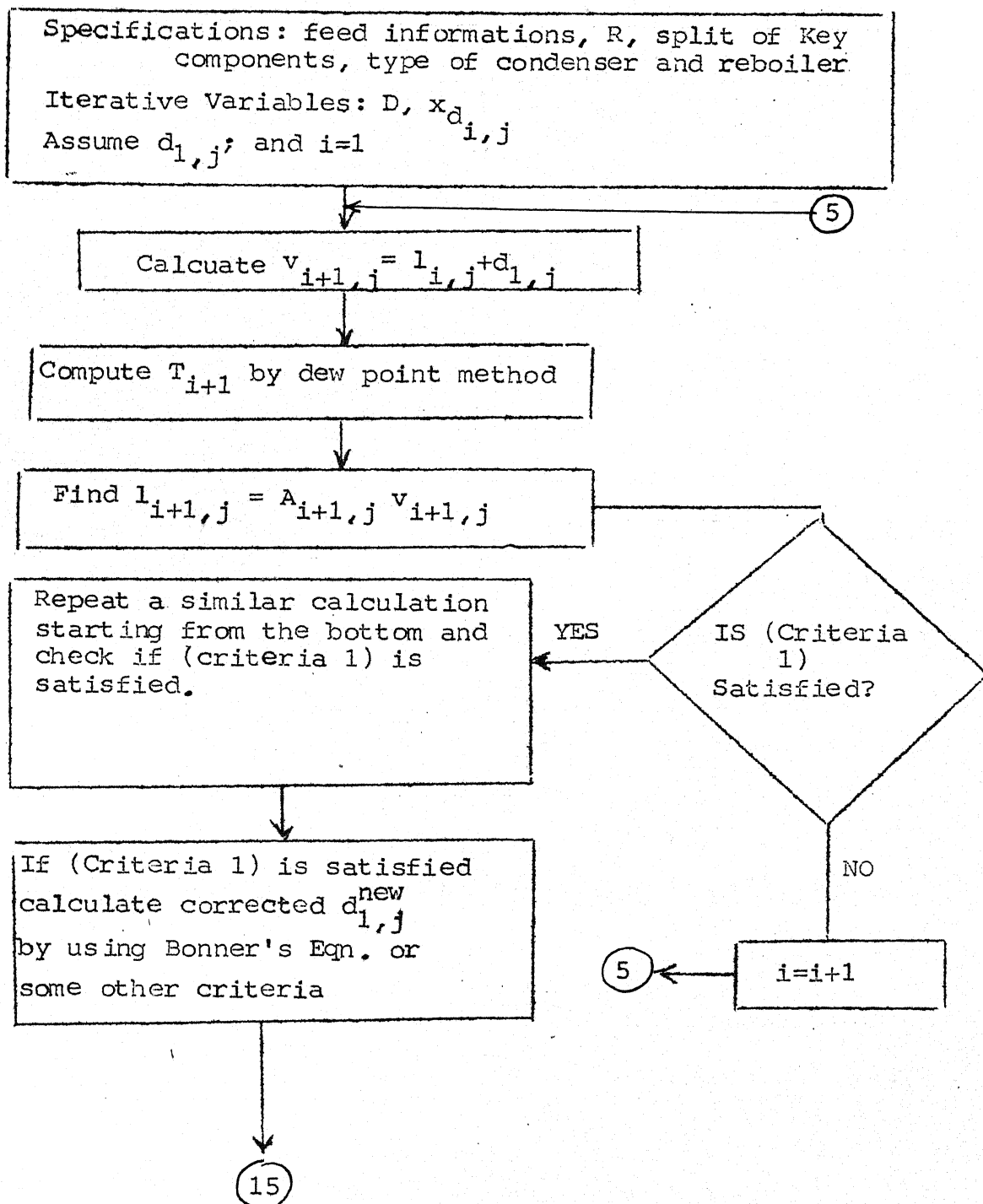
Underwood, A.V.J.: J. Inst. Petrol. 32, 598 (1946).

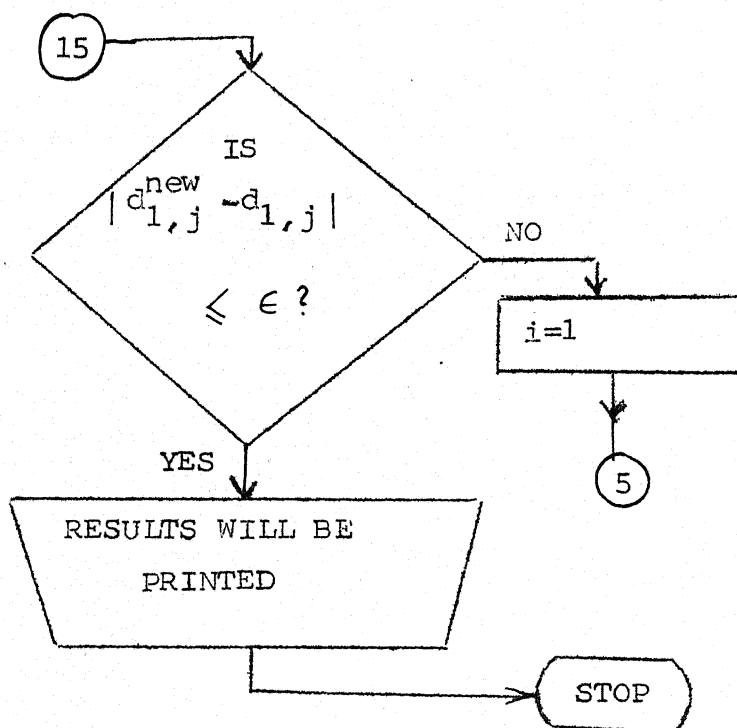
Underwood, A.V.J.: Chem. Eng. Prog., 44, 603 (1948).

Yen, L.C. and Alexander, R.E., AIChE J., 11, 334 (1965).

Zel'vansky, Y.D.: Zhurnal Prikladnoi Khimii, 54(1), 302 (1971)

(By courtesy Defence Scientific Information and Documentation Centre, Delhi-110054).

APPENDIX AFlow Chart for Lewis-Matheson Tray-by-Tray Algorithm (1932)



Note:

Criteria 1: $\frac{x_{i, lk}}{x_{i, hk}} \sim \frac{x_{F, lk}}{x_{F, hk}}$

where

lk = light Key and hk = heavy key component

Bonner's Eqn.:
$$d_{1,j}^{new} = \frac{(x_{f,j})_{B \rightarrow F} - (x_{f,j})_{D \rightarrow F}}{(x_{f,j}/d_{1,j})_{D \rightarrow F} - (x_{f,j}/b_{N,j})_{B \rightarrow F}}$$

APPENDIX B

Flowchart for the Ricker-Grens DSA Approach (1974)

Specifications: Feed information, key component split, R,D,P,
 type of condenser and reboiler
 Assume: T_i , $l_{i,j}$, $v_{i,j}$, N_E and N_S

⑤

Inner loop of Naphtali-Sandholm iterative
 solution to obtain corrected T_i , $l_{i,j}$ and
 $v_{i,j}$

IS $|\sum_{i=1,2,\dots,N} y_{i,j} - 1| \leq 0.05$?

NO

⑤

YES

Estimate changes in $(N_E + N_S)$ and (N_E/N_S)

Change
 in N_E and
 N_S 0.5?

NO

Adjust
 N_E and N_S

Estimate new T_i ,
 $v_{i,j}$ and $l_{i,j}$
 by scaling

⑤

YES

Are
 all func-
 tions ≤ 0 ?

YES

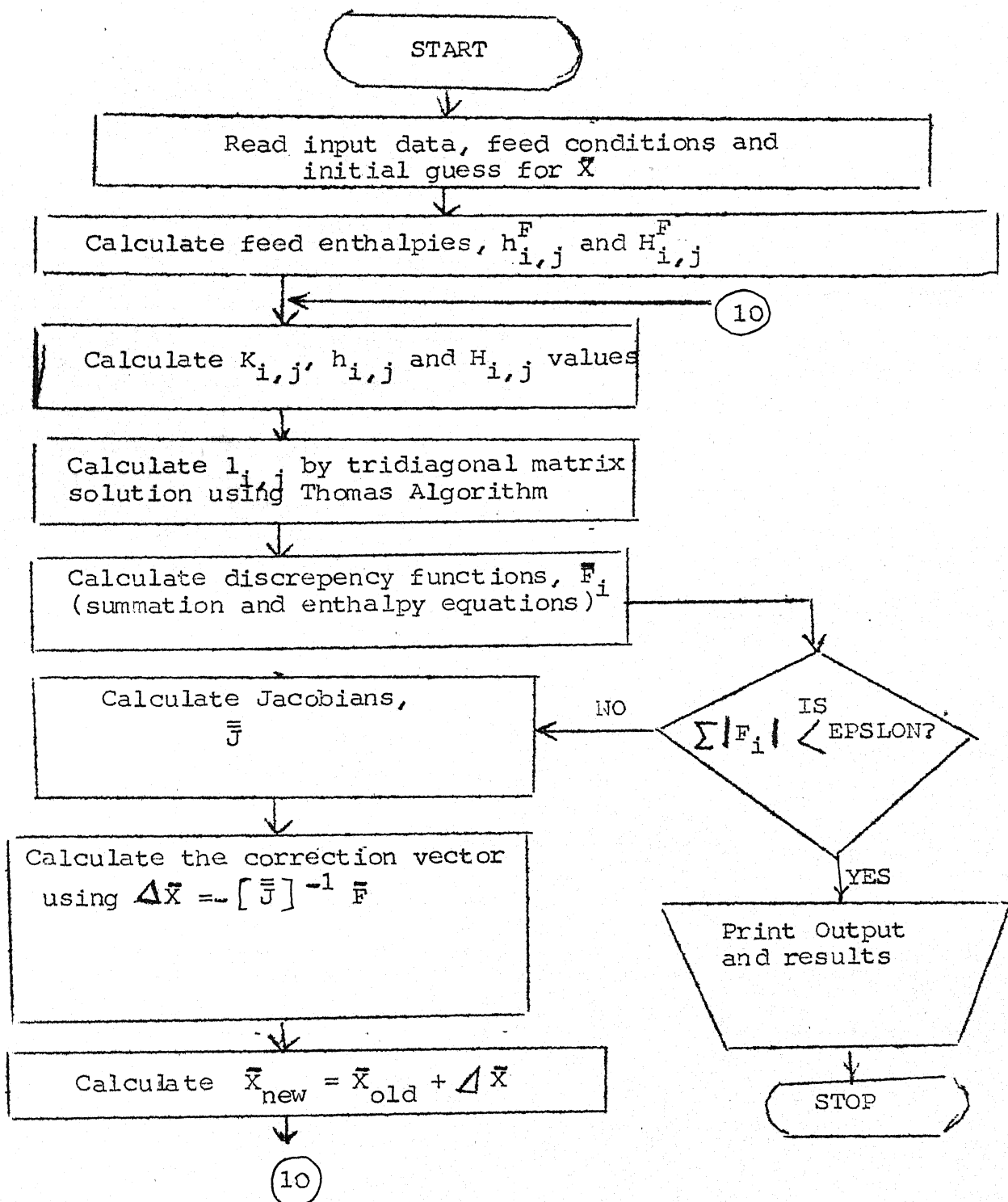
Print the results and
 output variables

STOP

NO

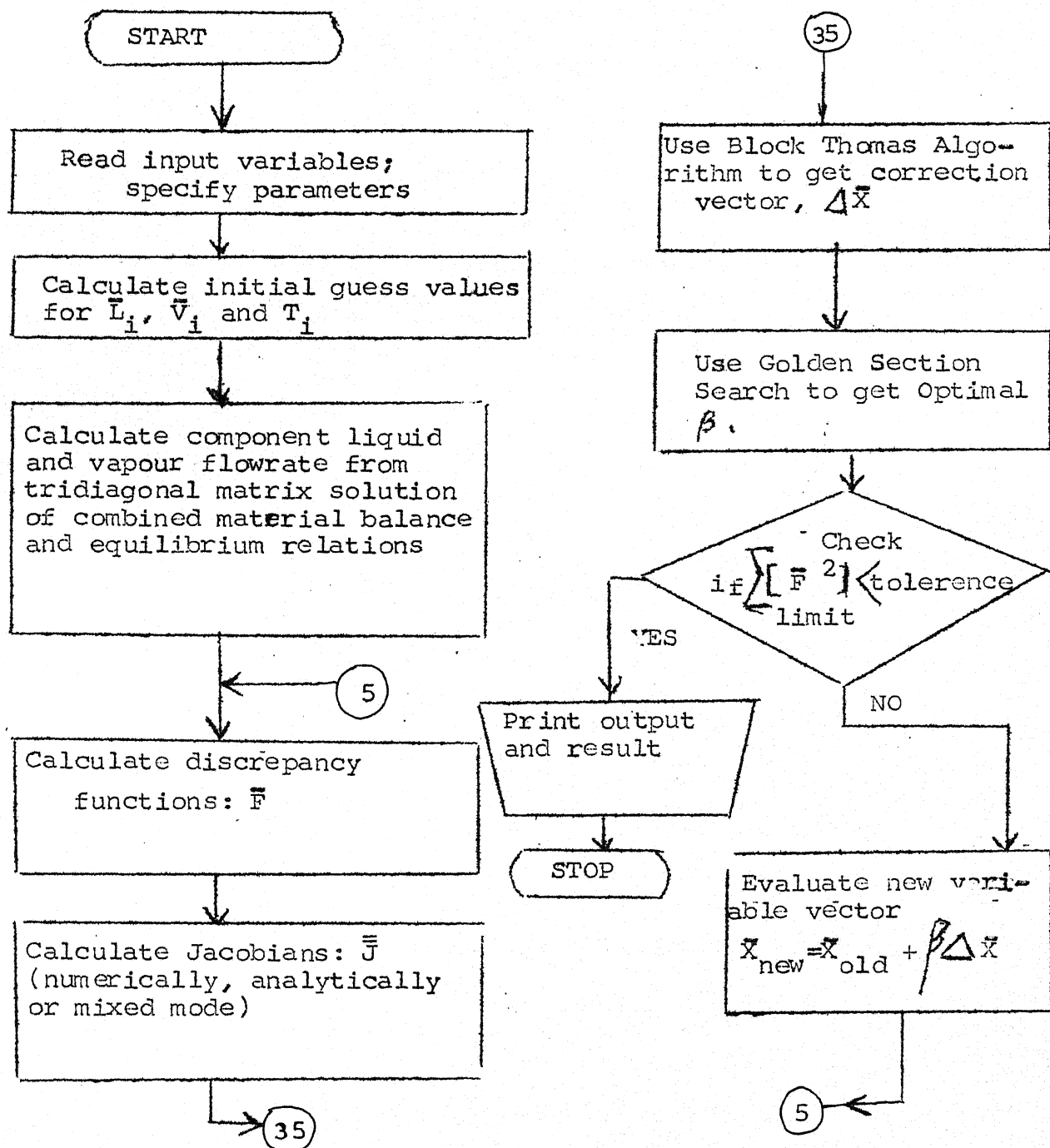
⑤

APPENDIX C

Flowchart for Ohmura Kasahara Algorithm (1978)

APPENDIX D

Simplified Flow Chart for the New Simultaneous Semi-tray-by-tray Algorithm (present method)



APPENDIX EMethod to Obtain Solution of Cubic Equation (Perry, 1963)

$$\text{Equation: } x^3 + bx^2 + cx + d = 0$$

$$p = \frac{1}{3} (3c - b^2)$$

$$q = \frac{1}{27} (27d^2 - 9bc + 2b^3)$$

$$R = (p/3)^3 + (q/2)^2$$

$$M = (-q/2 + \sqrt{R})^{1/3} \quad \text{positive value of } R \text{ used}$$

$$N = (-q/2 - \sqrt{R})^{1/3} \quad \text{-do-}$$

$$Y_1 = M + N$$

Roots

for

$R \geq 0$

$$Y_2 = \frac{1}{2} (M+N) + \frac{i\sqrt{3}}{2} (M-N)$$

$$Y_3 = \frac{1}{2} (M+N) - \frac{i\sqrt{3}}{2} (M-N)$$

$$x_i = Y_i - \frac{b}{3}; \quad i = 1, 2, 3$$

If $R > 0$, one real two complex conjugate roots

If $R = 0$, three real roots, of which at least two are equal

If $R < 0$, above formula are impractical, then use

$$x_i = \left(\sqrt[3]{-p/3} \right) \cos \left[\frac{\phi}{3} + 120^\circ K \right] - \frac{b}{3}, \quad K = 0, 1, 2$$

Roots

for

$R < 0$

$$\phi = \cos^{-1} \left(\frac{-q^2/4}{-p^3/27} \right), \quad \text{positive root taken}$$

Use upper (-) sign for $q > 0$, lower (+) sign if $q < 0$

APPENDIX :F

```

=====
*****
* DISTILLATION SIMULATION CALCULATION ::: SEMI-TRAY-BY-TRAY
* APPROACH OR NAPHTHALI-SANDHOLM APPROACH IS USED.
*****
* DISTILLATION DESIGN CALCULATION ::: NEW SIMULTANEOUS
* SEMI-TRAY-BY-TRAY APPROACH
*
* BY :::
* MR. S. GANGULY &
* MR. A. K. DAS &
* DR. D. M. SARAF [ THESIS SUPERVISOR ]
*****

```

MAIN LINE PROGRAM :::

1
2
3
4

```

DIMENSION T(20),FL(25),FV(25),V(25),RL(25),AA(25,5),
RK(25,5),AT(20),BT(20),CT(20),F(25),RESULT(25),
RVC(20,5),RLC(20,5),XX(25,25),A(25,25),B(25,25),
C(25,150),D(25,60),G(25),FLC(25,5),FVC(25,5),ENTL(25,5),
ENTV(25,5)
DIMENSION XXC(25,10),XXD(25,10),GD(25,10),GC(25,10),GT(25,25)
DIMENSION DR(25,5),PIE(25,5),XM(12)
DIMENSION SPRIME(2,25,5),SL(20),SV(20),EFI(25,5)
DIMENSION RLM(25),VM(25),ZX(25,5),KSEO(15)
DIMENSION XXXC(20,10),XXXD(20,10)
DIMENSION SDERIV(2,25,5),DERTV(25,5),XETAX(20,3,3),XETAY(20,3,3)
DIMENSION CPL(25,5),CPV(25,5)
DIMENSION ENTLF(25),ENTVF(25)
COMMON /AREA1/RK,M,P/AREA2/ENTV/AREA3/ENTL/AREA5/A,B,C,D,G
COMMON /AREA4/NJ,IPRINT/AREA7/FLC,FVC,REFLUX,DTS,RL,V,FL,FV
COMMON /AREA12/SUMMM/AREA13/GT/AREA14/ITN
COMMON /AREA16/XM/AREA17/IND/AR20/DR/AREA21/AA,SPRIME/Y1/PIE,ZX
COMMON /AREA10/EFI,SL,SV/AREA61/INDM
COMMON /AREA11/DERTV/AREA24/SDERTV/AREA25/RLM,VM
COMMON /AREA28/LOCK/AREA29/NUMBER/AREA50/KSEO
COMMON /AREA33/NUMBER,KTHERM/AREA8/CPL/AREA9/CPV
COMMON /AREA35/XX
COMMON /AREA37/XETAX,XETAY/AREA81/ICOND
COMMON /AREA30/ENTLF,ENTVF/AREA80/N

```

OPEN(UNIT=22,FILE='OUT.REF')

INPUT INFORMATIONS ::

DATA IPRINT,NJ,BETA,EPSLON,ITMAX/1,22,1,.01,8/
LOCK ----- INDICATES WHETHER SIMULATION OR DESIGN PROBLEM :::
IF LOCK=1 :: DESIGN WILL BE DONE
IF LOCK=2 :: SIMULATION WILL BE DONE

DATA LOCK/1/

DATA NUMBER/1/

KRY=1

NUMBER ----- DERIVATIVE ESTIMATION CONTROL PARAMETER IN
----- SIMULATION PROBLEM :::

IF 'NUMBER' = 1 : NUMERICAL DERIVATIVE CALCULATION

= 2 : ANALYTICAL DERIVATIVE CALCULATION

CAUTION :: ALWAYS SUPPLY NUMBER=2 FOR DESIGN MODE (I.e. LOCK=1)

DATA NUMBER/2/

KTHERM ----- PARAMETER INDICATING WHETHER IDEAL OR NON-IDEAL
THERMODYNAMICS IS TO BE USED :::

IF KPTHERM = 1 : SYSTEM IS IDEAL

= 2 : SYSTEM IS NONIDEAL

= 3 : SYSTEM NONIDEAL W.R.T EQUILIBRIUM ONLY

= 4 : SYSTEM NONIDEAL W.R.T EQUILIBRIUM & LIQUID ENTHALPY
I.e. VAPOUR PHASE IDEAL

DATA KTHERM/1/

IPRINT ----- PRINTING CONTROL PARAMETER ::::

IF IPRINT=1 : INPUT DATA WILL BE PRINTED

DATA IPRINT/3/

ICOND ----- CONDENSER CHOICE PARAMETER -----

= 1 : PARTIAL CONDENSER WITH VAP. STREAM ONLY AS PRODUCT

= 2 : PARTIAL CONDENSER WITH BOTH STREAMS AS PRODUCT

= 3 : TOTAL CONDENSER

DATA ICOND/3/

IF(LOCK.EQ.2) GO TO 8

----- DATA FOR DESIGN PROBLEM ::: -----

N=11

```

7313      FV(I,J)=0.0;EFT(I,J)=1.0
      CONTINUE
      FL(NFEED)=100.0
      FLC(NFEED,1)=23.0;FLC(NFEED,2)=37.0;FLC(NFEED,3)=40.0
      REFLUX=5.0;DIS=22.6;SC=1.0;SB=1.0
      KSEO(3)=2 ; KSEO(9)=2
      DO 10 J=9
C----- DATA FOR SIMULATION PROBLEM ::: -----
      CONTINUE
      NFEED=8;TFEED=225.0
      N=15
      M=3
      P=250.0 ; T(1)=120.0 ; T(N)=280.0
      DO 8313 I=1,N
      FL(I)=0.0 ; FV(I)=0.0
      SL(I)=0.0 ; SV(I)=0.0
      KSEO(I)=1.0
      DO 8313 J=1,M
8313      FLC(I,J)=0.0 ; FVC(I,J)=0.0 ; EFT(I,J)=1.0
      CONTINUE
      FL(NFEED)=100.0
      FLC(NFEED,1)=23.0;FLC(NFEED,2)=37.0 ; FLC(NFEED,3)=40.0
      REFLUX=5.0 ; DIS=22.6 ; SC=1.0 ; SB=1.0

      THE FOLLOWING DATA ARE TO BE SUPPLIED IN SUBROUTINES.:::
      -----
      EFT,SL,SV,ENTLF,ENTVF :: IN "GENG" SUBROUTINE
      MAP, ENTHALPY DATA :: IN "ENTV" SUBROUTINE
      LIO, ENTHALPY DATA :: IN "ENTL" SUBROUTINE
      KDERIV :: IN "GENB" SUBROUTINE
      EOLU, DATA :: IN "EOLBM" SUBROUTINE
      EPSDER :: IN "DERIV" SUBROUTINE
      -----
      CONTINUE
      IF(IPRINT.NE.1) GO TO 7249
      WRITE(22,7715)
      FORMAT(5X,40(1H-)/10X,'INPUT INFORMATION TO BE SUPPLIED :::')
      1 40(1H-)/)
      WRITE(22,7716) N,M,NFEED,REFLUX,DIS,P,SC,SB
      7715
      7716
      FORMAT(3X,'N',M,' OF STAGES =',I8/5X,'N',M,' OF COMPONENTS =',I8/
      1 5X,'FEED STAGE NO. =',I8/5X,'REFLUX RATIO =',E15.7/5X,'
      2 TOTAL DISTILLATE RATE =',E15.7/5X,'COLUMN PRESSURE =',E15.7
      3 5X,'SC=',F8.3,5X,'SB=',F8.3)
      7249
      CONTINUE
      DO 9542 I=1,N
      DO 9542 J=1,M
      DO 9542 KL=1,M
      YETAX(I,I,KL)=0.0
      YETAY(I,I,KL)=0.0
9542
      CONTINUE
      SECOND STEP
      ASSUME INITIAL VALUES OF RL,V,T BASED ON FEED CONDITION
      AND ASSUMING CONSTANT MOLAL OVERFLOW

      KKK=2*M+1
      INITIAL TEM. PROFILE :T:
      RATIO=(T(N)-T(1))/(N-1)
      DO 10 I=2,N-1
      IT=I-1
      T(I)=T(IT)+RATIO
      CONTINUE
      INITIAL GAS FLOWRATES :V:
      V(1)=DIS
      V(2)=DIS*(1.0+REFLUX)
      DO 15 I=3,N
      IT=I-1
      V(I)=V(IT)-FV(IT)
      CONTINUE
      RL(1)=REFLUX*DIS
      DO 20 I=2,N
      IT=I-1
      RL(I)=RL(IT)+FL(IT)
      CONTINUE
      IF(IPRINT) 21,21,22
      21
      WRITE(N3,1000)
      1000
      FORMAT(5X,'INITIAL GUESS VALUES - SECOND STEP')
      DO 24 I=1,N
      WRITE(N3,1111) RL(I),V(I),T(I),I
      1111
      FORMAT(5X,'RL(I) =',F10.5,4X,'V(I) =',F10.5,4X,'T(I) =',F10.5
      4X,'I =',I4)
      CONTINUE
      CONTINUE

```

```

C      COMPONENT FLOWRATES
C      ABSORPTION FACTORS:-
      DO 25 I=1,N
      FI=I(I)
      IB=I
      CALL EOLBM(TI,IB)
25     CONTINUE
      DO 30 I=1,N
      DO 35 J=1,M
      AA(I,J)=RL(I)/(V(I)*RK(I,J))
35     CONTINUE
30     CONTINUE
C      DEFF. MATRICES
      DO 40 J=1,M
      AA(1,J)=REFLUX
      DO 45 I=1,N
      TI=I-1
      BT(I)=-(AA(I,J)+1.0)
      IF ((I.EQ.N).AND.(SB.NE.1.0)) AA(N,J)=(SB/RK(N,J))
      IF(I.EQ.N) GO TO 42
      CT(I)=1.0
42     CONTINUE
      IF(I.EQ.1) GO TO 43
      AT(I)=AA(TI,J)
43     CONTINUE
      F(I)=-(FVCK(I,J)+FLC(I,J))
45     CONTINUE
      CALL THOMAS(N,AT,BT,CT,F,RESULT)
      LINE=16200
      TYPE*,LINE
C      COM. VAP. & LIO. FLOWRATES
      DO 46 I=1,N
      RVC(I,J)=RESULT(I)
      RLC(I,J)=AA(I,J)*RVC(I,J)
46     CONTINUE
40     CONTINUE
C      TEMPERATURE PROFILE IS SAME AS ASSUMED BEFORE
C      -----
C      ASSIGNMENT FOR XX VECTOR
C      -----
      DO 48 I=1,N
      DO 49 J=1,M
      XX(I,J)=RLC(I,J)
      KK=4+J
      XX(I,KK)=RVC(I,J)
49     CONTINUE
      XX(I,KKK)=T(I)
      IF(CLOCK.EQ.2) GO TO 48
      IF(KSEQ(I).EQ.2)XX(I,KKK)=2.0
48     CONTINUE
      LINE=18350
      TYPE*,LINE
C      -----
C      FEED ENTHALPY GENERATION STARTS :::::
C      -----
      IF(CLOCK.NE.30) GO TO 4159
      WRITE(22,56)
56     FORMAT(10X,'FEED ENTHALPY IS CALCULATED TO BE :::: /5X,60(1H*)/
      1 5X,'PLATE NO.',8X,'FOR LIQUID',8X,'FOR VAPOUR'/5X,9(1H-),8X,
      2 10(1H-),8X,10(1H-)//)
4159    CONTINUE
      DO 61 I=1,N
      IF(I.NE.IFEED) GO TO 62
      FI=I-EE
      IB=I
      IF(KTHERM.EQ.2) GO TO 71
      CALL ENTLIO(IB,FI,M)
      GO TO 73
71     CONTINUE
      CALL NNTIO(M,XX,IB,3,P,TI,RK)
73     SUM1=0.0
      DO 63 I=1,M
      SUM1=SUM1+(ENTL(I,J)*FLC(I,J))
63     CONTINUE
      ENTLF(I)=SUM1/FL(I)
      GO TO 61
62     ENTLF(I)=0.0 ; ENTVF(I)=0.0
61     CONTINUE
      DO 68 I=1,N
      WRITE(22,57) I,ENTLF(I),ENTVF(I)
57     FORMAT(8X,I3,8X,E15.8,6X,E15.8)
68     CONTINUE
C      ----- FEED ENTHALPY GENERATION COMPLETED -----
C      -----
      ITN=0.0
70     CONTINUE
C      ***** NEW ITERATION STARTED *****
C      ***** HERE *****
      LINE=21300
      TYPE*,LINE

```

```

IF(NUMDER.EO.2) GO TO 34
DO 33 I=1,N
IF(LOCK.EO.2) GO TO 331
IF(KSEO(I).EO.2) GO TO 33
331 CONTINUE
IF(J.EO.1) GO TO 5147
C *****CAUTION FOR PAR. COND. *****
PI=XX(I,KKK)
IB=I
CALL NUME08(TI,IB,XX)
5147 CONTINUE
IF(KTHERM.EO.2) GO TO 33
CALL NUMENT(TI,IB,M)
33 CONTINUE
TYPE *, ITN
IF(LPRINT.NE.2) GO TO 34
WRITE(22,2149)
2149 FORMAT(2X,'STAGE',3X,'COMP',4X,'XETAX(I,J)',6X,'XETAY(I,J)',6X,
1 'DERIV(I,J)/2X,5(1H-),3X,5(1H-),4X,10(1H-),5X,10(1H-),6X,
2 10(1H-))
DO 2154 IK=1,N
DO 2151 IM=1,M
DO 2151 IN=1,M
WRITE(22,2150) IK,IM,IN,XETAX(IK,IM,IN),XETAY(IK,IM,IN)
2150 1 'DERIV(IK,IM)
2151 FORMAT(5X,I3,3X,I3,4X,I3,2X,E12.6,5X,E12.6,5X,E12.6)
2151 CONTINUE
2154 CONTINUE
34 CONTINUE
ITN=ITN+1
TYPE *, ITN
SUMMM=0.0
DO 50 I=1,N
SUM=0.0
SUMM=0.0
DO 52 J=1,M
KK=M+J
SUMM=SUMM+XX(I,J)
SUM=SUM+XX(I,KK)
52 CONTINUE
RL(I)=SUMM
V(I)=SUM
50 CONTINUE
DO 51 I=1,N
IF(LOCK.EO.2) GO TO 57
IF(KSEO(I).EO.2) GO TO 6000
57 CONTINUE
PI=XX(I,KKK)
IB=I
IF(KTHERM.EO.2) GO TO 59
CALL ENTVAP(IB,PI,M)
CALL ENTLLC(IB,PI,M)
IF(KTHERM.EO.3) GO TO 59
CALL EOLBM(PI,IB)
59 CONTINUE
NOZ=3
IF(KTHERM.EO.3) NOZ=1
DO 6001 JL=1,NOZ
JIL=JL
CALL NONTOL(M,XX,IB,JIL,P,PI,RK)
6001 CONTINUE
GO TO 51
6000 CONTINUE
51 CONTINUE
DO 587 I=1,N
IF(KSEO(I).NE.2) GO TO 587
IND=1
DO 6010 IA=1,M
XM(IA)=XX(I,M+IA)/V(I)
6010 CONTINUE
X1=XX(I-1,KKK)
X2=XX(I+1,KKK)
IH=I
CALL REGF(X1,X2,0.0001,20,TCAL,IH)
PI=TCAL
6011 WRITE(05,6011) ITN,IND,TCAL
FORMAT(5X,'ITN=',I4,3X,'IND=',I4,3X,'TCAL=',F8.3)
CALL ENTVAP(IH,PI,M)
IF(KTHERM.EO.3) GO TO 339
CALL EOLBM(PI,IH)
GO TO 379
339 CONTINUE
NOZ=3
IF(KTHERM.EO.3) NOZ=1
DO 9914 JL=1,NOZ
JIL=JL
CALL NONTOL(M,XX,IH,JIL,P,PI,RK)
9914 CONTINUE

```

```

379      CONTINUE
      DO 6012 IA=1,M
      SDERIV(1,I,IA)=DERIV(1,IA)
6012      CONTINUE
      RRMJ=XX(I,KKK)
      WRITE(22,5678) RRMJ,T,ITN
5678      FORMAT(2X,'M=',E15.7,2X,'I=',I3,2X,'ITN=',I3)
      CALL ENVLIO(I+N,TI,M)
      X1=RL(I-1);X2=RL(I)
      INDM=3
      CALL REGF2(.005,20,X1,X2,I,XX,VALUE,N,M,RL,V)
      RLM(I)=VALUE
      DO 6020 IA=1,M
      SPRIME(1,I,IA)=(RK(I,IA)*(1.0+SV(I))*V(I))/RLM(I)
6020      CONTINUE
      IND=2
      X1=XX(I-1,KKK);X2=XX(I+1,KKK)
      DO 6030 IA=1,M
      XM(IA)=XX(I,IA)/RL(I)
6030      CONTINUE
      IB=I
      CALL REGF(X1,X2,0.0001,20,TCAL,IB)
      TI=TCAL
      WRITE(05,6031)ITN,IND,TCAL
6031      FORMAT(5X,'ITN=',I3,3X,'IND=',I3,4X,'TCAL=',F8.3)
      CALL ENVLIO(IB,TI,M)

      IF(KTHERM.EQ.3) GO TO 9812
      CALL EOLRM(TI,IB)
      GO TO 9813
9812      CONTINUE
      NOZ=3
      IF(KTHERM.EQ.3)NOZ=1
      DO 8914 JL=1,NOZ
      JIL=JL
      CALL NONIDL(M,XX,IB,IJL,P,TI,RK)
8914      CONTINUE
9813      CONTINUE
      CALL ENTVAP(I+N+1,TI,M)
      X1=V(I);X2=V(I+1);INDM=4
      CALL REGF2(.005,20,X1,X2,I,XX,VALUE,N,M,RL,V)
      VM(I)=VALUE
      WRITE(22,2541) RLM(I),VM(I),T
2541      FORMAT(2X,'RLM=',E15.7,2X,'VM=',E15.7,2X,'T=',I3)
      DO 6035 IA=1,M
      SDERIV(2,I,IA)=DERIV(1,IA)
6035      CONTINUE
      DO 6040 IA=1,M
      SPRIME(2,I,IA)=RK(I,IA)*VM(I)/((1.0+SL(I))*RL(I))
      AA(1,IA)=SORT((SPRIME(1,I,IA)*(1.0+SPRIME(2,I,IA)))+0.25)-0.5
      PERM=(AA(1,IA))*((RRMJ+1))
      DR(1,IA)=(AA(1,IA)-TERM)
      QIF(I,IA)=(1.0-TERM)
      ZX(1,IA)=(SPRIME(1,I,IA)*(SPRIME(2,I,IA)+1.0))/(SPRIME(1,I,IA)+
      1.0)
6040      CONTINUE
587      CONTINUE
      LINE=39400
      TYPE=LINE
      IF(IPRINT.NE.2) GO TO 7582
      DO 7582 I=1,M
      DO 7582 J=1,KKK
      WRITE(22,7581) XX(I,J),I,J
7581      FORMAT(2X,'XX=',E15.7,'I=',I3,2X,'J=',I3)
7582      CONTINUE

-----
C      COMPIATION OF DELTA.XX USING BLOCK THDMAS ALGORITHM::
C      -----
      IF(IPRINT.NE.2) GO TO 5569
      WRITE(22,5563) ITN
5563      FORMAT(17X,'ITERATION NO. = ',I3/2X,'[ DELTAX & GT GIVES
      1. PREVIOUS ITERATION VALUES :: 1'/2X,60(1H-))
      DO 5566 I=1,N
      DO 5566 J=1,KKK
      WRITE(22,5567) XX(I,J),C(J,I),I,J,GT(I,J)
5567      FORMAT(2X,'XX=',E15.7,5X,'DELTAX=',E15.7,2X,'I=',I3,2X,'J=',I3/
      1. 5X,'GT(I,J)='E15.7)
5566      CONTINUE
5569      CONTINUE
      CALL BTHDM(XX,KKK,N)
      GO TO 5247
      DO 3214 I=1,N
      DO 3214 J=1,KKK
      WRITE(22,3215)J,I,C(J,I)
3215      FORMAT(25X,'C('I2,',',I2,',)=',E15.7)
3214      CONTINUE
5247      CONTINUE
      GO TO 5249

```

```

(1,1)= 0.2140883E+02
(2,1)= 0.2642692E+00
(3,1)= 0.1088198E+01
(4,1)= 0.4281766E+01
(5,1)= 0.5285385E+01
(6,1)= 0.2176395E+02
(7,1)= 0.0000000E+00
(1,2)= 0.1758648E+02
(2,2)= 0.1220214E+01
(3,2)= -0.8193765E+02
(4,2)= 0.2569060E+02
(5,2)= 0.3171231E+00
(6,2)= 0.1305837E+01
(7,2)= 0.2389353E+01
(1,3)= -0.1704743E+02
(2,3)= 0.1518589E+02
(3,3)= 0.2284197E+01
(4,3)= 0.2186825E+02
(5,3)= 0.1273068E+01
(6,3)= -0.6017367E+02
(7,3)= 0.3020666E+01
(1,4)= -0.2196182E+02
(2,4)= 0.1464539E+02
(3,4)= 0.4662402E+01
(4,4)= -0.1276567E+02
(5,4)= 0.1523874E+02
(6,4)= 0.2286374E+01
(7,4)= 0.2052678E+02
(1,5)= -0.2591663E+02
(2,5)= 0.7765320E+01
(3,5)= -0.5442295E+01
(4,5)= -0.1768005E+02
(5,5)= 0.1469824E+02
(6,5)= 0.4664578E+01
(7,5)= -0.2252640E+02
(1,6)= -0.4274494E+02
(2,6)= -0.2635724E+01
(3,6)= 0.1205557E+01
(4,6)= -0.2163487E+02
(5,6)= 0.7818173E+01
(6,6)= 0.5444472E+01
(7,6)= 0.2774456E+02
(1,7)= -0.3384628E+02
(2,7)= -0.1024062E+01
(3,7)= -0.1460495E+01
(4,7)= -0.3846317E+02
(5,7)= -0.2582871E+01
(6,7)= 0.1207733E+01
(7,7)= -0.2225401E+02
(1,8)= -0.1949420E+02
(2,8)= 0.8393427E+01
(3,8)= 0.1877155E+01
(4,8)= -0.2956451E+02
(5,8)= -0.9712110E+00
(6,8)= 0.1462672E+01
(7,8)= 0.1140958E+02
(1,9)= -0.9248168E+01
(2,9)= 0.1336421E+02
(3,9)= 0.5501162E+01
(4,9)= -0.1521244E+02
(5,9)= 0.8446277E+01
(6,9)= 0.1879331E+01
(7,9)= -0.9824688E+00
(1,10)= 0.4994343E+00
(2,10)= 0.3295610E+02
(3,10)= 0.2181921E+02
(4,10)= -0.4966404E+01
(5,10)= 0.1341706E+02
(6,10)= 0.5503338E+01
(7,10)= -0.6041237E+01
(1,11)= -0.4281763E+01
(2,11)= 0.5284953E+01
(3,11)= -0.2174537E+02
(4,11)= 0.4781198E+01
(5,11)= 0.3300895E+02
(6,11)= 0.2182138E+02
(7,11)= -0.5315418E+01

```

5249

C
C

```

CONTINUE
NEW VALUES OF VARIABLES::
CHECK IN CONVERGENCE BEGINS
DO 110 I=1,N
DO 110 J=1,KKK
IF(ABS(CX(J,I)).GE..005) GO TO 120
110 CONTINUE
GO TO 999
C -
GOLDEN SECTION SEARCH BEGINS
120 CONTINUE
WRITE(05,5112)

```

110

C -

120

```

5112  FORMAT(15X,'GOLDEN SECTION SEARCH BEGINS _____')
      GGG=(SORT(5.0)-1.0)/2.0
      AAA=-1.0
      BBB=1.0
      INEW=1
113   DIFF=(BBB-AAA)
      IF(DIFF.LE..001) GO TO 143
      CCC=AAA+GGG*DIFF
      DDD=BBB-GGG*DIFF
      DO 140 I=1,N
      DO 140 J=1,KKK
      XXC(I,J)=XX(I,J)+CCC*C(J,I)
      IF(LOCK.EQ.2) GO TO 116
      IF(XXC(3,7).LT.2.0) XXC(3,7)=2.0
      IF(XXC(9,7).LT.2.0) XXC(9,7)=2.0
116   CONTINUE
      IF(XXC(I,J).LE.0.0) GO TO 137
      XXD(I,J)=XX(I,J)+DDD*C(J,I)
      IF(LOCK.EQ.2) GO TO 119
      IF(XXD(3,7).LT.2.0) XXD(3,7)=2.0
      IF(XXD(9,7).LT.2.0) XXD(9,7)=2.0
119   CONTINUE
      IF(XXD(I,J).LE.0.0) GO TO 138
140   CONTINUE
      GO TO 135
137   BBB=.9*BBB
      INEW=2
      GO TO 113
138   AAA=.9*AAA
      INEW=2
      GO TO 113
135   CONTINUE
      DO 9113 I=1,N
      DO 9113 J=1,KKK
      XXXC(I,J)=XXC(I,J)
      XXXD(I,J)=XXD(I,J)
9113  CONTINUE
      CALL FUN(XXC,GC,N)
      IF(LFN.NE.30) GO TO 9001
      WRITE(05,9002)
9002  FORMAT(5X,'CCC---CALLING ENDS:::')
9001  CONTINUE
      CCCDUM=CCC
      DDDDUM=DDD
      CALL FUN(XXD,GD,N)
      IF(LFN.NE.30) GO TO 9003
      WRITE(05,9004)
9004  FORMAT(5X,'DDD---CALLING ENDS:::')
9003  CONTINUE
      SUMFC=0.0
      SUMFD=0.0
      DO 150 I=1,N
      DO 150 J=1,KKK
      SUMFC=SUMFC+ABS(GC(I,J))
      SUMFD=SUMFD+ABS(GD(I,J))
150   CONTINUE
      IF(LFN.NE.30) GO TO 8889
      WRITE(05,8888) CCC,SUMFC,DDD,SUMFD
8888  FORMAT(2X,'CCC=',F10.5,2X,'SUMFC=',F15.6,3X,'DDD=',F10.5
      1X,2X,'SUMFD=',F15.6)
8889  CONTINUE
      IF(SUMFC.LT.SUMFD) GO TO 153
      IF(SUMFC.GT.SUMFD) GO TO 163
      IF(SUMFC.EQ.SUMFD) GO TO 173
153   AAA=DDD
      INEW=1
      GO TO 113
163   BBB=CCC
      INEW=1
      GO TO 113
173   AAA=DDD
      BBB=CCC
      INEW=1
      GO TO 113
143   IF(SUMFC.LE.SUMFD) GO TO 144
      IF(SUMFC.GT.SUMFD) GO TO 146
144   RSUMF=SUMFC
      BETA=CCC
      IF(INEW.EQ.2) BETA=CCCDUM
      DO 145 I=1,N
      DO 145 J=1,KKK
      XX(I,J)=XXC(I,J)
      IF(INEW.EQ.2) XX(I,J)=XXXC(I,J)
      GT(I,J)=GC(I,J)
145   CONTINUE
      GO TO 148
146   DO 147 I=1,N
      DO 147 J=1,KKK
      XX(I,J)=XXD(I,J)

```

```

147 IF(INEW.EQ.2) XX(I,J)=XXXD(I,J)
      CT(I,J)=G(I,J)
      CONTINUE
      BETA=DDD
      IF(INEW.EQ.2) BETA=DDDDUM
      RSUMF=SUMFD
148 *IF(05,5060) SUMMM,RSUMF,BETA
5060 FORMAT(2X,'OLD SUM',E11.5,2X,'NEW SUM',E11.5,2X,
      'DPT.FRAC',E11.5)
      IF(ITN.GE.ITMAX) GO TO 250
      IF(RSUMF.LT..01) GO TO 999

      IF(RSUMF.LT.SUMMM) GO TO 70
      IF(RSUMF.GE.SUMMM) GO TO 170.
-----
CHECK FOR CONVERGENCE CRITERIA.
-----
999 CONTINUE
      DO 90 I=1,N
      DO 95 J=1,KKK
      WRITE(ND,2000) XX(I,J),C(J,I),I,J
      FORMAT(2X,'XX(I,J)=',E15.7,2X,E15.7,2X,'I=',I4,2X,'J=',I4)
      WRITE(22,2005) GT(I,J),I,J
      FORMAT(2X,'GT=',E11.5,2X,'I=',I3,2X,'J=',I3)
2005 CONTINUE
95 CONTINUE
90 CONTINUE
      IF(ABL.EQ.1) GO TO 795
      GO TO 260
170 WRITE(ND,5020)
5020 FORMAT(2X,'CONVERGENCE IS NOT ACHETVED')
      IBL=1
      GO TO 999
250 WRITE(ND,3000) ITN
3000 FORMAT(5X,'ITN=',I4/2X,'ITN EQUALS ITMAX-PROGRAM TERMINATED')
      GO TO 999
260 CONTINUE
      IF(ITN.GE.ITMAX) GO TO 795
      WRITE(22,798) ITN
798 FORMAT(2X,'CONVERGENCE IS ACHETVED IN ',I3,' ITERATIONS')
795 CONTINUE
      DO 9901 I=1,N
      WRITE(22,9757) I,XX(I,2*M+1)
9757 FORMAT(9X,I2,10X,F9.4)
      DO 9901 J=1,M
      WRITE(22,9889) J,XX(I,J),XX(I,J+M)
9889 FORMAT(15X,I2,2X,E12.6,3X,E12.6)
9901 CONTINUE
      STOP
      END
      MAIN OVER IS OVER
      *****

```

```
*****
REGF : : :
*****
```

```
SUBROUTINE REGF(X,Y,EPS,ITMAX,RESULT,NJ)
```

```
COMMON /AREA17/IND
```

```
ITNP=0
```

```
L=1
```

```
CONTINUE
```

```
IF(IND.EQ.2) GO TO 110
```

```
CALL FUNC(X,FX,NJ)
```

```
CALL FUNC(Y,FY,NJ)
```

```
GO TO 120
```

```
CONTINUE
```

```
CALL FUNC(X,FX,NJ)
```

```
CALL FUNC(Y,FY,NJ)
```

```
CONTINUE
```

```
Z=((X*FY)-(Y*FX))/(FY-FX)
```

```
IF(IND.EQ.1) CALL FUNC(Z,FZ,NJ)
```

```
IF(IND.EQ.2) CALL FUNC(Z,FZ,NJ)
```

```
IF((ABS(FZ).LT.EPS)) GO TO 50
```

```
IF(ITNP.GT.ITMAX) GO TO 60
```

```
ITNP=ITNP+1
```

```
IF(L.EQ.1) GO TO 30
```

```
IF(L.EQ.2) GO TO 35
```

```
CONTINUE
```

```
Y=Z
```

```
L=L-1
```

```
GO TO 80
```

```
CONTINUE
```

```
X=Z
```

```
L=L+1
```

```
GO TO 80
```

```
WRITE(05,70)
```

```
FORMAT(2X,'REGULA-FALSI IS NOT CONVERGING IN 20 ITERATIONS')
```

```
CONTINUE
```

```
RESULT=Z
```

```
RETURN
```

```
END
```

```
*****
```

```
*****
```

```
FUNC : : :
```

```
*****
```

```
SUBROUTINE FUNC(T,FX,J)
```

```
DIMENSION XM(12),RK(25,5),X(5),Y(5),Z(5),XN(5),YN(5)
```

```
COMMON /AREA15/XM /AREA1/RK,M,P/AREA33/NUMBER,KTHERM
```

```
IF(KTHERM.NE.1) GO TO 50
```

```
SUM=0.0
```

```
CALL EOLRM(T,J)
```

```
DO 20 I=1,M
```

```
SUM=SUM+(XM(I)/RK(J,I))
```

```
CONTINUE
```

```
FX=SUM-1.0
```

```
RETURN
```

```
CONTINUE
```

```
V=M;S=.5;EPSI=.001;ICOUNT=0;PT=P
```

```
DO 100 I=1,M
```

```
Z(I)=XM(I)
```

```
Y(I)=Z(I)
```

```
X(I)=1.0/V
```

```
CONTINUE
```

```
CONTINUE
```

```
IF(ICOUNT.NE.40) GO TO 8547
```

```
WRITE(05,935) ICOUNT
```

```
FORMAT(2X,'ICOUNT=',I3)
```

```
CONTINUE
```

```
IF(ICOUNT.EQ.41) GO TO 84
```

```
XSUM=0.0
```

```
CALL NIDL2(M,X,Y,J,1,P,T,RK)
```

```
DO 44 I=1,M
```

```
XN(I)=Y(I)/RK(J,I)
```

```
XSUM=XSUM+XN(I)
```

```
CONTINUE
```

```
DO 54 I=1,M
```

```
XN(I)=XN(I)/XSUM
```

```
XN(I)=(1.0-S)*X(I)+S*XN(I)
```

```
CONTINUE
```

```
DO 251 I=1,M
```

```
IF(ABS(XN(I)-X(I)).GT.EPSI) GO TO 252
```

```
CONTINUE
```

```
GO TO 255
```

```
CONTINUE
```

```
DO 253 I=1,M
```

```
X(I)=XN(I)
```

```
CONTINUE
```

```

      ICOUNT=ICOUNT+1
      GO TO 34
255  CONTINUE
      XSUM=0.0
      DO 256 I=1,N
      XSUM=XSUM+Z(I)/RK(J,I)
256  CONTINUE
      FX=1.0-XSUM
      FX=-FX
      RETURN
84  WRITE(05,552)
552  FORMAT(2X,'X-LOOP IS NOT CONVERGING IN 20 ITERATIONS')
      STOP
      END
C*****
C*****
      FUNG :::
C*****8
      SUBROUTINE FUNG(T,FX,J)
      DIMENSION XM(12),RK(25,5),X(5),Y(5),Z(5),XN(5),YN(5)
      COMMON /AREA16/XM/AREA1/RK,M,P/AREA33/NUMBER,KTHERM
      IF(KTHERM.NE.1) GO TO 50
      SUMD=0.0
      CALL EOLBM(T,J)
      DO 20 I=1,M
      SUMD=SUMD+XM(I)*RK(J,I)
20  CONTINUE
      FX=SUMD-1.0
      RETURN
50  CONTINUE
      N=M;S=.6;EPSI=.001;ICOUNT=0;PT=P
      DO 100 I=1,N
      Z(I)=XM(I)
      X(I)=Z(I)
      Y(I)=1.0/N
100  CONTINUE
34  CONTINUE
      IF(ICOUNT.NE.40) GO TO 8548
      WRITE(05,935) ICOUNT
935  FORMAT(2X,'ICOUNT=',I3)
8548  CONTINUE
      IF(ICOUNT.EQ.41) GO TO 84
      XSUM=0.0
      CALL NIDL2(M,X,Y,J,1,P,T,RK)
      DO 44 I=1,N
      YN(I)=X(I)*RK(J,I)
      XSUM=XSUM+YN(I)
44  CONTINUE
      DO 54 I=1,M
      YN(I)=YN(I)/XSUM
      YN(I)=(1.0-S)*Y(I)+S*YN(I)
54  CONTINUE
      DO 251 I=1,N
      IF(ABS(YN(I)-Y(I)).GT.EPSI) GO TO 252
251  CONTINUE
      GO TO 255
252  CONTINUE
      DO 253 I=1,N
      Y(I)=YN(I)
253  CONTINUE
      ICOUNT=ICOUNT+1
      GO TO 34
255  CONTINUE
      XSUM=0.0
      DO 256 I=1,N
      XSUM=XSUM+Z(I)*RK(J,I)
256  CONTINUE
      FX=XSUM-1.0
      RETURN
84  WRITE(05,552)
552  FORMAT(2X,'Y LOOP IS NOT CONVERGING IN 41 ITERATIONS')
      STOP
      END
C*****
C*****
      SUBROUTINE FOR THERMODYNAMIC PROPERTY ESTIMATION :::
C*****
      SUBROUTINE NIDL2(NV,X,Y,JSP,INDEX,PTZ,TZ,RK)
      DIMENSION PC(5),TC(5),ZC(5),OMEGA(5),VC(5),X(5),Y(5),KI(5)
      DIMENSION CPVAPA(5),CPVAPB(5),CPVAPC(5),CPVAPD(5)
      DIMENSION DELHV(10),YDUM(10)
      DIMENSION ENTL(25,5),ENTV(25,5),CPL(25,5),CPV(25,5)
      DIMENSION RK(25,5),XX(25,25)
      COMMON /MADAT/N,NCOMP,KSITU,NR,NPRI
      COMMON /SYSDAT/PC,TC,ZC,OMEGA,VC
      COMMON /AREA2/ENTL/AREA9/CPV/AREA3/ENTL/AREA8/CPL
      REAL KI
      DATA KEY/2/
      DATA NCOMP,KSITU,NR,NPRI/3,3,0,0/

```

```

C ----- IDEAL GAS HEAT CAPACITY EQUATION CONSTANTS ----:
DATA (CPVAPA(K),K=1,3)/-1.009,2.266,-0.866/
DATA (CPVAPB(K),K=1,3)/7.315E-02,7.913E-02,1.164E-01/
DATA (CPVAPC(K),K=1,3)/-3.789E-05,-2.647E-05,-5.613E-05/
DATA (CPVAPD(K),K=1,3)/7.978E-09,-0.674E-09,1.267E-08/
C ----- MOLECULAR DATA FOR NONIDEAL THERMODYNAMICS ----:
DATA (PC(K),K=1,3)/41.9,37.5,33.3/
DATA (TC(K),K=1,3)/369.8,425.2,469.6/
DATA (ZC(K),K=1,3)/0.281,0.274,0.262/
DATA (OMEGA(K),K=1,3)/0.152,0.274,0.251/
C ----- IF INDEX=2 : VAPOUR ENTHALPY IS CALCULATED
INDEX=1 : EQUILIBRIUM CALCULATION IS DONE
INDEX=3 : LIQUID ENTHALPY IS ONLY CALCULATED

```

```

V=NV
PT=PTZ/14.7
T=((PTZ-32.0)*(5.0/9.0))+273.15
IF(INDEX.GE.2) GO TO 275
IF(JSP.EQ.1) RETURN
CALL EQUIL(T,PT,X,Y,KI)
DO 129 K=1,N
RK(JSP,K)=KI(K)
CONTINUE
129 IF(NPRI.EQ.0) GO TO 500
IF(KSITU.EQ.1) GO TO 20
WRITE(05,10)
20 FORMAT(15X,'EOLBM CONST BY PENG-ROBINSON EON. ')
GO TO 30
25 WRITE(05,25)
25 FORMAT(15X,'EOLBM CONST BY SAOAVE-REDLICH-KWONG EON. ')
30 CONTINUE
DO 40 NN=1,N
WRITE(05,35) JSP,KI(NN),NN
35 FORMAT(5X,'JSP=',I3,2X,'KI(N)=',E16.8,4X,'FOR COMP. NO. = ',I3)
40 CONTINUE
GO TO 500
275 CONTINUE
IF(INDEX.EQ.3) GO TO 278
IF(JSP.EQ.1) RETURN
DO 300 I=1,N
DO 276 LL=1,N
YDUM(LL)=0.0
IF(LL.EQ.00) YDUM(I)=1.0
276 CONTINUE
CALL ZDOT(ZMAX,ZMIN,YDUM,F.B.FM,TC,PC,T,PT,OMEGA,Z,N,KSITU)
PCDAT=PC(I)
TCDAT=TC(I)
ZCDAT=ZC(I)
CPA=CPVAPA(I)
CPB=CPVAPB(I)
CPC=CPVAPC(I)
CPD=CPVAPD(I)
CALL ENTHLP(PT,I,Z,TCDAT,PCDAT,ZCDAT,CPA,CPB,CPC,CPD,HACT,DELH,
1 CPVDAT)
ENTV(JSP,I)=CPVDAT
ENTV(JSP,I)=HACT
DELHV(I)=DELH
IF(KEY.EQ.0) GO TO 300
WRITE(05,280) JSP,I,ENTV(JSP,I),DELHV(I),CPV(JSP,I)
280 FORMAT(5X,I4,3X,'COMP.NO=',I3,3X,'HV=',E15.8,3X,'DELHV=',E15.8,
1 3X,'CPV=',E15.8)
300 CONTINUE
GO TO 500
278 CONTINUE

```

```

C ----- LIQUID ENTHALPY CALCULATION STARTS USING
C EQUATION OF CORRESPONDING STATES ----:

```

```

DO 310 I=1,N
OMDAT=OMEGA(I)
PCDAT=PC(I)
TCDAT=TC(I)
CPA=CPVAPA(I)
CPB=CPVAPB(I)
CPC=CPVAPC(I)
CPD=CPVAPD(I)
CALL ENTL2(TCDAT,PCDAT,T,PT,CPA,CPB,CPC,CPD,OMDAT,CPLDAT,DELHL,
1 DELHVT)
ENTL(JSP,I)=DELHL
CPL(JSP,I)=CPLDAT
IF(KEY.EQ.0) GO TO 310
WRITE(05,293) JSP,I,ENTL(JSP,I),DELHVT,CPL(JSP,I)
293 FORMAT(5X,I4,3X,'COMP.NO=',I3,3X,'HL=',E15.8,3X,'DELHVT=',E15.8,
1 3X,'CPL=',E15.8)
310 CONTINUE
500 CONTINUE
RETURN
END

```

```

C *****
C *****

```

REGF2 ::

SUBROUTINE REGF2(EPS,ITMAX,X,Y,I,XX,VALUE,N,M,RL,V)

DIMENSION XX(25,25),XM(12),RL(25),V(25)

COMMON /AREA16/XM/AREA61/INDM

VJ=N+1

ITNP=0

L=1

CONTINUE

IF(INDM.EQ.4) GO TO 110

CALL FUND(X,FX,NJ,XX,I,N,M,RL,V)

CALL FUND(Y,FY,NJ,XX,I,N,M,RL,V)

GO TO 120

CONTINUE

CALL FUNR(X,FX,NJ,XX,I,N,M,RL,V)

CALL FUNR(Y,FY,NJ,XX,I,N,M,RL,V)

CONTINUE

Z=((X*FY)-(Y*FX))/(FY-FX)

IF(INDM.EQ.3) CALL FUND(Z,FZ,NJ,XX,I,N,M,RL,V)

IF(INDM.EQ.4) CALL FUNR(Z,FZ,NJ,XX,I,N,M,RL,V)

IF((ABS(FZ).GT.EPS)) GO TO 50

IF(ITNP.GT.ITMAX) GO TO 60

ITNP=ITNP+1

IF(L.EQ.1) GO TO 30

IF(L.EQ.2) GO TO 35

CONTINUE

V=Z

L=L-1

GO TO 80

CONTINUE

X=Z

L=L+1

GO TO 80

WRITE(22,70)

FORMAT(2X,'REGULA-FALSI2 IS NOT CONVERGING IN 20 ITERATIONS')

CONTINUE

VALUE=Z

RETURN

END

FUND ::

SUBROUTINE FUND(X,FX,NJ,XX,I,N,M,RL,V)

DIMENSION XX(25,25),RL(25),V(25),XM(12),RK(25,5)

1.,ENTL(25,5),ENTV(25,5)

COMMON /AREA61/INDM/AREA17/IND/AREA1/RK,M,P/AREA2/ENTV

1./AREA3/ENTL/AREA16/XM

SUMA=0.0

DO 10 IA=1,M

XX(NJ,IA)=X*(XX(I,IA+M)/V(I))/RK(I,IA)

XX(NJ,IA+M)=XX(NJ,IA)+XX(I,IA+M)-XX(I-1,IA)

SUMA=SUMA+XX(NJ,IA+M)

CONTINUE

DO 20 IA=1,M

YA(IA)=XX(NJ,IA+M)/SUMA

CONTINUE

KKK=2*M+1

IND=1:X3=XX(I-1,KKK):X4=XX(I+1,KKK)

CALL REGF(X3,X4,.005,20,RES,NJ)

INT=RES

CALL ENTVAP(NJ,INT,M)

SUMJ=0.0

DO 30 IA=1,M

L=IA+4

SUMJ=SUMJ+(XX(I-1,IA)*ENTL(I-1,IA)+XX(NJ,L)*ENTV(NJ,IA))

1.-(XX(I,L)*ENTV(I,IA)+XX(NJ,IA)*ENTL(NJ,IA))

CONTINUE

FX=SUMJ

RETURN

END

FUNR :::

SUBROUTINE FUNR(X,FX,NJ,XX,I,N,M,RL,V)

DIMENSION XX(25,25),RL(25),V(25),XM(12),RK(25,5)

1.,ENTL(25,5),ENTV(25,5)

COMMON /AREA61/INDM/AREA17/IND/AREA1/RK,M,P/AREA2/ENTV

1./AREA3/ENTL/AREA16/XM

SUMA=0.0

DO 10 IA=1,M

XX(NJ+1,IA+M)=X*(RK(I,IA)*(XX(I,IA)/RL(I)))

XX(NJ+1,IA)=(XX(I,IA)+XX(NJ+1,IA+M))-XX(I+1,IA+M)

SUMA=SUMA+XX(NJ+1,IA)

CONTINUE

DO 20 IA=1,M

XM(IA)=XX(NJ+1,IA)/SUMA

CONTINUE

RETURN

END

```

20  CONTINUE
   KKK=2*M+1
   IND=2; X3=XX(I-1, KKK); X4=XX(I+1, KKK)
   CALL REGF(X3, X4, .005, 20, RES, NJ+1)
   TNJ1=RES
   CALL ENTLIO(NJ+1, TNJ1, M)
   SUMJ=0.0
   DO 30 IA=1, M
   SUMJ=SUMJ+(XX(NJ+1, IA)*ENTL(NJ+1, IA)+XX(I+1, IA+M)*ENTV(I+1, IA))
   1  -(XX(NJ+1, IA+M)*ENTV(NJ+1, IA)+XX(I, IA)*ENTL(I, IA))
30  CONTINUE
   SX=SUMJ
   RETURN
   END
*****

```

```

*****
GENX ::
*****8
SUBROUTINE GENX(J,N)
  DIMENSION X(25,25),Y(25,25)
  COMMON /AREA6/X,Y
  DO 20 IA=1,N
  DO 20 IB=1,N
  IF (IA.EQ.IB) EPS=0.0
  IF (IA.NE.IB) EPS=0.0
  X(IA,IB)=EPS
  CONTINUE
  RETURN
END
*****
GENY ::
*****
SUBROUTINE GENY(J,N)
  DIMENSION Y(25,25),X(25,25)
  COMMON /AREA6/X,Y
  DO 25 IA=1,N
  DO 25 IB=1,N
  IF (IA.EQ.IB) EP=0.0
  IF (IA.NE.IB) EP=0.0
  Y(IA,IB)=EP
  CONTINUE
  RETURN
END
*****
GENG::
*****
GENG : RETURNS THE VALUES OF DISCREPANCY FUNCTIONS
SUBROUTINE GENG(J,NJ,XX)
  DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25,
  1 X(25,25),Y(25,25),XX(25,25),FLC(25,5),FVC(25,5),EFI(25,
  2 5),SL(20),SV(20),RK(25,5),RL(25),V(25),ENTL(25,5)
  3 ,ENTV(25,5),ENTLF(25),ENTVF(25),FL(25),FV(25)
  DIMENSION XM(12),ZX(25,5),KSEO(15)
  DIMENSION DR(25,5),PTE(25,5)
  COMMON /AREA4/NJ,IPRINT/AREA5/A,B,C,D,G/AREA6/X,Y/I
  1 AREA7/FLC,FVC,REFLUX,DIS,RT,V,FL,FV/AREA1/RK,M,P/
  2 AREA2/ENTV/AREA3/ENTL/AREA12/SUMMM/AREA10/EFI,SL,SV
  COMMON /AREA14/ITN/AREA16/XM/AREA17/IND
  COMMON /AR20/DR/Y1/PTE,ZX/AREA50/KSEO
  COMMON /AREA28/LOCK/AREA30/ENTLF,ENTVF
  SUM1=0.0
  SUM=0.0
  IF((J.NE.1).AND.(J.NE.NJ)) GO TO 53
  IF(J.EQ.NJ) GO TO 30
  DO 50 IA=1,M
  L=IA+1
  G(IA)=-(1.0+SL(1))*XX(1,IA)-(1.0+SV(1))*XX(1,L)+XX(2,L)
  G(L)=XX(1,IA)-REFLUX*XX(1,L)
  CONTINUE
  G(N)=XX(1,N)-120.0
  GO TO 85
  CONTINUE
  KTER=2*M+1
  PTE=XX(NJ,KTER)
  DO 60 IA=1,M
  L=IA+1
  JJ=J-1
  G(IA)=XX(JJ,IA)-(1.0+SL(JJ))*XX(J,IA)-(1.0+SV(JJ))*XX(J,L)
  G(L)=(RK(J,IA)*XX(J,IA))/RL(J)-XX(J,L)/V(J)
  IF(IPRINT.NE.30) GO TO 8519
  WRITE(22,8517) NJ,IA,RK(NJ,IA),RL(NJ),V(NJ),XX(NJ,IA),XX(NJ,L)
  1 ,G(L)
  8517 FORMAT(2X,I3,2X,I3,'RK=',E15.7,2X,'RL=',E15.7,2X,'V=',E15.7/
  1 2X,'XXLT0=',E15.7,2X,'XXVAP=',E15.7,'G=',E15.7)
  8519 CONTINUE
  60 CONTINUE
  G(N)=RL(NJ)-77.4
  IF(LOCK.EQ.1) G(N)=RL(NJ)-77.4
  GO TO 85
  CONTINUE
  53 DO 10 IA=1,M
  G(1)=->G(M) :: COM. MAT. BALANCE
  G(M+1)->G(2M) :: EQU. RELATIONS
  G(2M+1)->ENTHALPY BALANCE
  JJ=J+1
  L=IA+M
  JJ=J-1
  G(IA)=XX(JJ,IA)-(1.0+SL(JJ))*XX(J,IA)-(1.0+SV(JJ))*XX(J,L)

```

```

IF(LJCK.EQ.2) GO TO 1511
IF(KSEO(J).EQ.2) GO TO 1500
CONTINUE
G(L)=(EFI(J,IA)*RK(J,IA)*XX(J,IA))/RL(J)-XX(J,L)/V(J)+
1 (1.0-EFI(J,IA))*(XX(JJ,L)/V(JJJ))
GO TO 2000
1500 CONTINUE
G(L)=((XX(JJ,IA)-XX(J,IA))/XX(JJ,IA))*PIE(J,IA)-
1 (1.0-(XX(J,IA)-XX(JJ,IA)+XX(J,L))/(XX(JJ,IA)*ZX(T,TA)))
2 *DR(J,IA)
G(L)=G(L)*XX(JJ,IA)
2000 CONTINUE
SUM=SUM+XX(JJ,IA)*ENTL(JJ,IA)-(1.0+SL(J))*(ENTL(J,IA)
1 *XX(J,IA))-(1.0+SV(J))*(ENTV(J,IA)*XX(J,L))+
2 (XX(JJ,L)*
10 ENTV(JJ,IA))
70 CONTINUE
CONTINUE
G(N)=SUM+FL(J)*ENTLF(J)+FV(J)*ENTVF(J)
IF(LOCK.EQ.2) GO TO 85
IF(J.EQ.3) G(4)=XX(3,4)-130.4214
85 IF(J.EQ.9) G(5)=XX(9,5)-61.33745
RETURN
END
SUBROUTINE GENG COMPLETED
*****
*****
*****
GENA::
*****
*****
*****
GENA:: RETURNS THE COEFFI. OF "A" MATRIX
SUBROUTINE GENA(J,N,NJ,XX)
1 DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25)
2 XX(25,25),SL(20),SV(20),EFI(25,5),RK(25,5),RL(25),
3 V(25),ENTL(25,5),ENTV(25,5),FLC(25,5),FVC(25,5)
,PL(25),FV(25),CPL(25,5),CPV(25,5),DERIV(25,5)
DIMENSION XM(12),SDERIV(2,25,5),ZX(25,5),KSEO(15)
DIMENSION DELTA(25,24),ALPHA(12,12)
DIMENSION DR(25,5),PIE(25,5),AA(25,5),SPRIME(2,25,5)
DIMENSION RLM(25),VM(25),ENTLF(25),ENTVF(25)
COMMON /AREA5/A,B,C,D,G/AREA10/EFI,SL,SV/AREA1/RK,M,P/AREA2/
1 ENTV/AREA3/ENTL/AREA9/CPV/AREA8/CPL/AREA7/FLC,FVC,REFLUX
2 ,DIS,RL,V,FL,FV/AREA11/DETRV/AREA16/XM/AREA17/IND
COMMON /AR20/DR/Y1/PTE,ZX/AREA21/AA,SPRIME
COMMON /AREA23/DELTA/AREA25/RLM,VM/AREA24/SDERIV
COMMON /AREA27/SUM1,SUM2/AREA28/LOCK/AREA29/NUMBER
COMMON /AREA30/ENTLF,ENTVF/AREA50/KSEO
LINE=13200;TYPE*,LINE
IF(J.EQ.1) GO TO 50
IJ=J-1
ME=(N-1)/2
DO 10 IA=1,M
DO 20 IB=1,N
DEL=0.0
IF(IA.EQ.IB) DEL=1.0
A(IA,IB)=DEL
IC=IA+M
A(IC,IB)=0.0
CONTINUE
CONTINUE
LINE=14400;TYPE*,LINE
IF(J.EQ.NJ) GO TO 777
SUM=0.0
SUM1=0.0
SUM2=0.0
DO 1010 IA=1,M
SUM1=SUM1+(XX(IJ-1,IA)*ENTL(IJ-1,IA))+(XX(IJ+M,IA)*ENTV(IJ+1,IA))
SUM2=SUM2+(XX(IJ,IA)*ENTL(IJ,IA))+(XX(IJ,IA+M)*ENTV(IJ,IA))
1010 CONTINUE
SUM1=SUM1+(FL(IJ)*ENTLF(IJ))+(FV(IJ)*ENTVF(IJ))
DO 30 IB=1,M
A(N,IB)=ENTL(IJ,IB)
IF(LOCK.EQ.2) GO TO 355
IF(KSEO(IJ-1).EQ.2) GO TO 300
GO TO 355
300 CONTINUE
SUM3=0.0
DO 310 IK=1,M
SUM3=SUM3+(XX(IJ-1,IK)*CPL(IJ-1,IK))
310 CONTINUE
TERM10=SUM3
A(N,IB)=A(N,IB)+(TERM10*DELTA(IJ-1,IB))
355 CONTINUE
ID=IB+M
A(N,ID)=0.0
SUM=SUM+XX(IJ,IB)*CPL(IJ,IB)
30 CONTINUE
LINE=17000;TYPE*,LINE
A(N,N)=SUM

```

```

IF(KSEQ(J),EO,2) GO TO 900.
IF(KSEQ(J-1),EO,2) A(N,N)=0.0
GO TO 50
777 CONTINUE
DO 450 IB=1,N
A(V,IB)=0.0
450 CONTINUE
GO TO 50
900 CONTINUE
IF(NUMBER,EO,1) GO TO 912
DO 800 IC=M+1,N-M
DO 800 IB=1,M
IA=IC-M
II=IA
TERM1=-0.5*(SPRIME(2,J,II)+1.0)*(SPRIME(1,J,II)/RLM(J))*(1.0-
1 (1.0/XX(J,N)))/(AA(J,II)+0.5)
TERM2=PIE(J,II)/AA(J,II)
TERM3=(1.0-AA(J,II))
TERM4=TERM3*(1.0-(XX(J,N)*TERM2))+AA(J,II)*(1.0-TERM2)
TERM5=TERM4/(TERM3*TERM3)
TERM6=(XX(J,II+M)*(1.0+SV(J))*TERM5*TERM1)-DR(J,II)-
1 (XX(J-1,II)*TERM5*TERM1)-((1.0+SL(J))*XX(J,II)*XX(J,N)*
2 TERM2*TERM1)
IF(IA,NE,IB) TERM6=TERM6+DR(J,II)
TERM6=TERM6/V(J)
A(IC,IB)=TERM6
800 CONTINUE
GO TO 50
912 CONTINUE
DO 926 IAD=1,M
KI=J-1
INDEX=1
JB=J:IADDUM=IAD
CALL JACOB2(XX,JB,N,M,IADDUM,INDEX,KI,ALPHA)
DO 926 IC=M+1,N-M
II=IC-M
A(IC,IAD)=ALPHA(II,IAD)
926 CONTINUE
KI=J-1:INDEX=3:IAD=N
CALL JACOB2(XX,J,N,M,IAD,INDEX,KI,ALPHA)
DO 939 IA=1,M
A(IA+M,N)=ALPHA(IA,N)
939 CONTINUE
RETURN
END
*****
GEND ::
*****
GEND :: RETURNS THE COEFF. OF "D" MATRIX
SUBROUTINE GEND(J,N,NJ,XX)
1 DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25)
2 ,XX(25,25),SL(20),SV(20),EFL(25,5),RK(25,5),RL(25),
3 V(25),ENTL(25,5),ENTV(25,5),FLC(25,5),FVC(25,5)
,FL(25),FV(25),CPL(25,5),CPV(25,5),DERIV(25,5)
DIMENSION X4(12),SDERIV(2,25,5)
DIMENSION DELTA(25,24),ALPHA(12,12)
DIMENSION DR(25,5),PIE(25,5),AA(25,5),SPRIME(2,25,5)
DIMENSION RLM(25),VM(25),ZX(25,5),KSEQ(15)
COMMON /AREA5/A,B,C,D,G/AREA10/EFL,SL,SV/AREA1/RK,M,R/AREA2/
1 ENTV/AREA3/ENTL/AREA9/CPV/AREA8/CPL/AREA7/FLC,FVC,REFLUX
2 ,DIS,RL,V,FL,FV/AREA11/DERIV/AREA16/XM/AREA17/IND
COMMON /AR20/DR/AREA21/AA,SPRIME/Y1/PIE,ZX/AREA50/KSEQ
COMMON /AREA23/DELTA/AREA25/RLM,VM/AREA24/SDERIV
COMMON /AREA27/SUM1,SUM2/AREA28/LOCK/AREA29/NUMBER
DATA IFLAG/1/
IF(J,EO,NJ) GO TO 50
M=(N-1)/2
IF(J,NE,1) GO TO 205
DO 200 IA=1,N
DO 200 IB=1,M
D(IA,IB)=0.0
200 CONTINUE
DO 210 IA=1,M
DO 210 IB=M+1,N
II=IB-M
DEL=0.0
IF(IA,EO,II) DEL=1.0
D(IA,IB)=DEL
210 CONTINUE
DO 215 IA=M+1,N-M
DO 215 IB=M+1,N
D(IA,IB)=0.0
215 CONTINUE
DO 223 IB=1,N
D(N,IB)=0.0
223 CONTINUE
GO TO 50
205 CONTINUE

```

```

20  DO 20 IB=1,M
10  D(IA,IB)=0.0
CONTINUE
CONTINUE
DO 30 IA=1,M
DO 40 IB=M+1,N
IT=IB-M
DEL=0.0
IF(IA.EQ.IC) DEL=1.0
D(IA,IB)=DEL
CONTINUE
CONTINUE
DO 60 IA=M+1,N-1
DO 70 IB=M+1,N-1
IT=IA-M
DEL=0.0
IF(LOCK.EQ.2) GO TO 553
IF(KSEQ(J).EQ.2) GO TO 550
CONTINUE
IF(IA.EQ.IB) DEL=(1.0-EFI(J,IT))*(V(J+1)-XX(J+1,IA))/
1 (V(J+1)*V(J+1))
IF(IA.NE.IB) DEL=(1.0-EFI(J,IT))*(-XX(J+1,IA)*(1.0/(V(J+1)
1 *V(J+1))))
GO TO 520
CONTINUE
IF(IFLAG.EQ.1) GO TO 520
TERM1=0.5*SPRIME(1,J,IT)*(SPRIME(2,J,IT)/VM(J))*(1.0-(1.0/
1 XX(J,N)))/(AA(J,IT)+0.5)
TERM2=PIE(J,IT)/AA(J,IT)
TERM3=(1.0-AA(J,IT))
TERM4=TERM3*(1.0-(XX(J,N)*TERM2))+(AA(J,IT)*(1.0-TERM2))
TERM5=TERM4/(TERM3*TERM3)
DEL=(XX(J,IA)*(1.0+SV(J))*TERM5*TERM1)-(XX(J-1,IT)*TERM5*TERM1)-
1 ((1.0+SL(J))*XX(J,IT)*XX(J,N)*TERM2*TERM1)
DEL=DEL/V(J)
D(IA,IB)=DEL
CONTINUE
D(IA,N)=0.0
CONTINUE
IF(LOCK.EQ.2) GO TO 497
IF(KSEQ(J).EQ.2) GO TO 451
GO TO 497
CONTINUE
IF(IFLAG.EQ.1) GO TO 453
GO TO 497
CONTINUE
DO 482 IAD=M+1,N-1
KJ=J+1
INDEX=2
JB=J+1;IADJUM=IAD
CALL JACOB2(XX,JB,N,M,IADJUM,INDEX,KJ,ALPHA)
DO 482 ICC=M+1,N-1
IIC=ICC-M
D(IIC,IAD)=ALPHA(IIC,IAD)
CONTINUE
INDEX=1;KJ=J+1;IAD=N
CALL JACOB2(XX,J,N,M,IAD,INDEX,KJ,ALPHA)
DO 529 IA=1,M
D(IA+M,N)=ALPHA(IA,N)
CONTINUE
CONTINUE
DO 80 IB=M+1,N-1
IT=IB-M
D(N,IB)=ENTV(J+1,IT)
IF(LOCK.EQ.2) GO TO 356
IF(KSEQ(J+1).EQ.2) GO TO 301
GO TO 356
CONTINUE
SUM3=0.0
DO 311 IK=1,M
SUM3=SUM3+(XX(J+1,IK+M)*CPV(J+1,IK))
CONTINUE
TERM11=SUM3
D(N,IB)=D(N,IB)+(TERM11*DELTA(J+1,IB))
CONTINUE
CONTINUE
SUM=0.0
DO 90 II=1,M
IA=II+M
SUM=SUM+XX(J+1,IA)*CPV(J+1,II)
CONTINUE
D(N,N)=SUM
IF(LOCK.EQ.2) GO TO 50
IF(KSEQ(J+1).EQ.2) D(N,N)=0.0
CONTINUE
RETURN
END
"D" MATRIX COMPLETED *****

```

```

GENB :::
*****

```

```

GENB :: RETURNS THE COEFF. OF "B" MATRIX

```

```

SUBROUTINE GENB(J,N,NJ,XX)
DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25)
1  ,XX(25,25),SL(20),SV(20),EFI(25,5),RK(25,5),RL(25),
2  V(25),ENTL(25,5),ENTV(25,5),FLC(25,5),FVC(25,5)
3  ,FL(25),FV(25),CPL(25,5),CPV(25,5),DERIV(25,5)
DIMENSION XM(12),SDERIV(2,25,5),XETAX(20,3,3),XETAY(20,3,3)
DIMENSION DELTA(25,24),RLM(25),VM(25)
DIMENSION DR(25,5),PIE(25,5),AA(25,5),SPRIME(2,25,5)
DIMENSION GAMMA(25,5),ALPHA(12,12)
DIMENSION ZX(25,5),KSEO(15)
COMMON /AREA29/NUMBER/AREA37/XETAX,XETAY
COMMON /AREA5/A,B,C,D,G/AREA10/EFI,SL,SV/AREA1/RK,M,P/AREA2/
1  ENTV/AREA3/ENTL/AREA9/CPV/AREA8/CPL/AREA7/FLC,FVC,REFLUX
2  ,DTS,RL,V,FL,FV/AREA11/DERIV/AREA16/XM/AREA17/TND
COMMON /AR20/DR/AREA21/AA,SPRIME/Y1/PIE,ZX/AREA50/KSEO
COMMON /AREA23/DELTA/AREA25/RLM,VM/AREA24/SDERIV
COMMON /AREA27/SUM1,SUM2/AREA28/LOCK
DATA KDERIV/1/
DO 10 IA=1,M
DO 20 IB=1,M
DEL=0.0
IF(IA.EQ.IB) DEL=-(1.0+SL(J))
BY(IA,IB)=DEL
CONTINUE
DO 30 IB=M+1,N
IC=IB-M
DEL=0.0
IF(IA.EQ.IC) DEL=-(1.0+SV(J))
BY(IA,IB)=DEL
CONTINUE
CONTINUE
IF(J.NE.1) GO TO 810
DO 820 IA=M+1,N-1
DO 820 IB=1,M
L=IB+M
DEL=0.0
IF(IA.EQ.L) DEL=1.0
BY(IA,IB)=DEL
DEL=0.0
IF(IA.EQ.L) DEL=-REFLUX
BY(IA,L)=DEL
CONTINUE
DO 830 IA=M+1,N-1
BY(IA,0)=0.0
CONTINUE
DO 840 IB=1,N-1
BY(M,IB)=0.0
CONTINUE
BY(M,N)=1.0
GO TO 110
810 CONTINUE
IF(KDERIV.EQ.1) GO TO 25
F=XX(J,N)
CALL KDERIV(J,T)
CONTINUE
25 IF(LOCK.EQ.2) GO TO 833
IF(KSEO(J).EQ.2) GO TO 600
833 CONTINUE
DO 40 IA=M+1,N-1
II=IA-M
DO 50 IB=1,M
DEL=0.0
IF(II.EQ.IB) DEL=(EFI(J,IB)*RK(J,IB))*(RL(J)-XX(J,IB))/
1  (RL(J)*RL(J))+(EFI(J,IB)*XX(J,IB)*XETAX(J,II,IB)/RL(J))
IF(II.NE.IB) DEL=(EFI(J,II)*RK(J,II))*(-XX(J,II))*(1.0/(RL
1  (J)+RL(J)))+(EFI(J,II)*XX(J,II)*XETAX(J,II,IB)/RL(J))
BY(IA,IB)=DEL
CONTINUE
CONTINUE
DO 45 IA=M+1,N-1
DO 60 IB=M+1,N-1
DEL=0.0
IF(IA.EQ.IB) DEL=-(V(J)-XX(J,IB))/(V(J)*V(J))+(EFI(J,IB-M)*XX(
1  J,IA-M)*XETAY(J,IA-M,IB-M)/RL(J))
IF(IA.NE.IB) DEL=XX(J,IA)/(V(J)*V(J))+(EFI(J,IA-M)*XX(J,
1  IA-M)*XETAY(J,IA-M,IB-M)/RL(J))
BY(IA,IB)=DEL
60 CONTINUE
II=IA-M
BY(IA,N)=(EFI(J,II)*XX(J,II)*DERIV(J,II))/RL(J)
45 CONTINUE
GO TO 900
600 CONTINUE
IF(NUMBER.EQ.1) GO TO 581

```

```

DD 620 IB=1,M
DEL=0.0
II=IA-M
TERM3=-0.5*SPRIME(1,J,II)*SPRIME(2,J,II)/(RL(J))
TERM8=0.5*SPRIME(1,J,II)*(VM(J)*SDERIV(2,J,II)*DELTA(J,IB)/
1 ((1.0+SL(J))*RL(J)))
TERM9=-0.5*(1.0+SPRIME(2,J,II))*SPRIME(1,J,II)*(1.0+SL(J))/
1 (XX(J,N)*RLM(J))
TERM2=(TERM3+TERM8+TERM9)/(AA(J,II)+0.5)
TERM4=-((1.0+SL(J))*((XX(J,II)*XX(J,N)*(PIE(J,II)/
1 AA(J,II))*TERM2))
TERM5=PIE(J,II)/AA(J,II)
TERM1=1.0-AA(J,II)
TERM5=(TERM1*(1.0-XX(J,N)*TERM6))+(AA(J,II)*(1.0-TERM6))
TERM7=TERM5/(TERM1*TERM1)
DEL=TERM4-(XX(J-1,II)*TERM7*TERM2)+((1.0+SV(J))*XX(J,IA)*
1 TERM7*TERM2)
IF(II.EQ.IB) DEL=DEL-(1.0+SL(J))*PIE(J,II)
DEL=DEL/V(J)
B(IA,IB)=DEL
CONTINUE
GO TO 597
581 CONTINUE
DO 582 IAD=1,M
KJ=J
INDEX=1
JB=J:IADDUM=IAD
CALL JACOB2(XX,JB,N,M,IADDUM,INDEX,KJ,ALPHA)
DO 582 IC=M+1,N-1
II=IC-M
B(IC,IAD)=ALPHA(II,IAD)
CONTINUE
CONTINUE
IF(NUMBER.EQ.1) GO TO 681
DO 710 IA=M+1,N-1
II=IA-M
DO 710 IB=M+1,N-1
DEL=0.0
TERM8=(SPRIME(1,J,II)/((1.0+SV(J))*V(J)))+(1.0+SV(J))*
1 V(J)*SDERIV(1,J,II)*DELTA(J,IB)/RLM(J))
TERM9=SPRIME(1,J,II)*SPRIME(2,J,II)*(1.0+SV(J))/(VM(J)*XX(J,N))
TERM1=0.5*(((SPRIME(2,J,II)+1.0)*(TERM8))+TERM9)*(1.0/
1 (AA(J,II)+0.5))
TERM2=PIE(J,II)/AA(J,II)
TERM3=1.0-AA(J,II)
TERM4=((TERM3*(1.0-XX(J,N)*TERM2))+(AA(J,II)*(1.0-TERM2)))*TERM1
TERM5=TERM4/(TERM3*TERM3)
TERM6=(1.0+SV(J))*(XX(J,IA)*TERM5)
TERM10=TERM6-(XX(J-1,II)*TERM5)-((1.0+SL(J))*XX(J,II)*XX(J,N)*
1 TERM2*TERM1)

```

```

TERM11=XX(1,IA)*(1.0+SV(J))*(1.0+DR(J,II))-(DR(J,II)*XX(J-1,II))
1 -(PIE(J,II)*(1.0+SL(J))*XX(J,II))
TERM12=TERM10+((1.0+SV(J))*(1.0+DR(J,II)))
IF(IA.EQ.IB) TERM12=TERM10
DEL=-TERM11/(V(J)*V(J))+(TERM12/V(J))
CONTINUE
B(IA,IB)=DEL
CONTINUE
GO TO 697
681 CONTINUE
DO 682 IAD=M+1,N-1
KJ=J
INDEX=2
JB=J:IADDUM=IAD
CALL JACOB2(XX,JB,N,M,IADDUM,INDEX,KJ,ALPHA)
DO 682 IC=M+1,N-1
II=IC-M
B(IC,IAD)=ALPHA(II,IAD)
CONTINUE
CONTINUE

```

```

DO 740 IA=M+1,N=1
II=IA-M
TERM1=-SPRIME(1,J,II)*(SPRIME(2,J,II)/VM(J))*(VM(J)-W(J+1))/
1 XX(J,N)
TERM2=(SPRIME(2,J,II)+1.0)*SPRIME(1,J,II)*(RLM(J)-RL(J-1))/
1 (RLM(J)*XX(J,N))
TERM3=0.5*(TERM1+TERM2)/(AA(J,II)+0.5)
TERM4=PIE(J,II)*(ALOG(AA(J,II))+(XX(J,N)*TERM3/AA(J,II)))
TERM5=(PIE(J,II)/AA(J,II))
TERM6=(1.0)/((1.0-AA(J,II))*2.0)
TERM7=(1.0-TERM5)*TERM6*TERM3
TERM8=-AA(J,II)/(1.0-AA(J,II))
TERM9=TERM5*(ALOG(AA(J,II)))+((XX(J,N)-1.0)*TERM3/AA(J,II))
TERM10=TERM7+(TERM8*TERM9)
B(IA,N)=(XX(J,IA)*(1.0+SV(J))*TERM10)-(XX(J-1,II)*TERM10)
1 -(1.0+SL(J))*XX(J,II)*TERM4)
B(IA,N)=B(IA,N)/V(J)
740 CONTINUE
DO 752 TO 900
752 CONTINUE
CALL JACO8(XX,J,N,GAMMA)
DO 753 IA=M+1,N=1
753 B(IA,N)=GAMMA(J,IA-M)
900 CONTINUE
CONTINUE
IF(J.EQ.NJ) GO TO 120
SUM=0.0
DO 70 IB=1,M
70 IT=IB+M
B(N,IB)=-((1.0+SL(J))*ENTL(J,IB))
IF(LOCK.EQ.2) GO TO 1050
IF(KSEQ(J).EQ.2) GO TO 1030
1030 GO TO 1050
CONTINUE
SUM3=0.0
DO 1020 IK=1,M
1020 SUM3=SUM3+(XX(J,IK)*CPL(J,IK))
CONTINUE
TERM13=SUM3
B(N,IB)=B(N,IB)-(TERM13*(1.0+SL(J))*DELTA(J,IB))
1050 CONTINUE
B(N,II)=-((1.0+SV(J))*ENTV(J,IB))
IF(LOCK.EQ.2) GO TO 1120
IF(KSEQ(J).EQ.2) GO TO 1110
1110 GO TO 1120
CONTINUE
SUM4=0.0
DO 1130 TK=1,M
1130 SUM4=SUM4+(XX(J,IK+M)*CPV(J,IK))
CONTINUE
TERM14=SUM4
B(N,II)=B(N,II)-(TERM14*(1.0+SV(J))*DELTA(J,II))
1120 CONTINUE
SUM=SUM-(1.0+SL(J))*XX(J,IB)*CPL(J,IB)-((1.0+SV(J))*XX(J,II)
*CPV(J,IB))
70 CONTINUE
B(N,N)=SUM
IF(LOCK.EQ.2) GO TO 110
IF(KSEQ(J).EQ.2) B(N,N)=0.0
120 GO TO 110
CONTINUE
DO 330 IB=1,M
330 B(N,IB)=1.0
IF(13.EQ.3) B(N,3)=1.0
IF(LOCK.EQ.2) B(N,IB)=1.0
I=IB+M
B(N,I)=0.0
CONTINUE
B(N,N)=0.0
110 RETURN
END
"8" MATRIX COMPLETED
*****
*****
GNGT' :::
*****
SUBROUTINE GNGT(J,N)
DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25),GT(25,25)
COMMON /AREA13/GT/AREA5/A,B,C,D,G/AREA14/ITN
DO 10 I=1,N
10 G(I)=GT(J,I)
CONTINUE
RETURN
END
*****

```

```

*****
FUN ::::
*****
SUBROUTINE FUN(XX,GT,N)
DIMENSION XX(25,25),GT(25,25),RK(25,5),FLC(25,5),FVC(25,5),
1 RL(25),V(25),FL(25),FV(25),A(25,25),B(25,25),C(25,150),D(25,60)
2 ,G(25),ENTV(25,5),ENTL(25,5)
DIMENSION DR(25,5),PIE(25,5),XM(12)
DIMENSION SPRIME(2,25,5),SL(20),SV(20),EFI(25,5)
DIMENSION AA(25,5),ZX(25,5),KSEO(15)
DIMENSION RLM(25),VM(25)
COMMON /AREA16/XM/AREA17/IND/AR20/DR/AREA21/AA,SPRIME/Y1/PIE,ZX
COMMON /AREA1/RK,M,P/AREA7/FLC,FVC,REFLUX,DIS,RL,V,FL,FV/
1 AREA5/A,B,C,D,G/AREA2/ENTV/AREA3/ENTL
COMMON /AREA10/EFI,SL,SV/AREA14/ITN
COMMON /AREA33/NUMBER,KTHERM
COMMON /AREA28/LOCK/AREA50/KSEO/AREA61/INDM
NIDEAL=3
KEY=100
IF(ITN.EQ.30)KEY=0
KKK=2*M+1
DO 50 I=1,N
SUM=0.0
SUMM=0.0
DO 52 J=1,M
KK=M+J
SUMM=SUMM+XX(I,J)
SUM=SUM+XX(I,KK)
52 CONTINUE
RL(I)=SUMM
V(I)=SUM
50 CONTINUE
DO 51 I=1,N
IF(LOCK.EQ.2) GO TO 5999
IF(KSEO(I).EQ.2) GO TO 6000
5999 CONTINUE
TI=XX(I,KKK)
TB=I
IF(KTHERM.EQ.2) GO TO 2813
CALL ENTVPAP(TB,TI,M)
CALL ENTLIO(TB,TI,M)
IF(KTHERM.EQ.3) GO TO 2813
CALL EOLBM(TI,TB)
2813 GO TO 51
CONTINUE
NOZ=3
IF(KTHERM.EQ.3) NOZ=1
DO 2816 K2=1,NOZ
KKZ=K2
2816 CALL NNTOL(M,XX,IB,KKZ,P,TI,RK)
CONTINUE
GO TO 51
6000 CONTINUE
51 CONTINUE
DO 5543 I=1,N
IF(KSEO(I).NE.2) GO TO 5543
IND=1
DO 6010 IA=1,M
XM(IA)=XX(I,M+IA)/V(I)
6010 CONTINUE
X1=XX(I-1,KKK)
X2=XX(I+1,KKK)
CALL REGF(X1,X2,0.0001,20,TCAL,I)
TI=TCAL
IF(KEY.NE.0) GO TO 8889
WRITE(22,8889) IND,TCAL
8889 FORMAT(2X,'IND=',I3.2X,'TCAL=',F10.5)
CONTINUE
IC=I
CALL ENTVPAP(IC,IT,M)
IF(KTHERM.NE.1) GO TO 7912
CALL EOLBM(TI,IC)
7912 GO TO 7913
CONTINUE
NOZ=3
IF(KTHERM.EQ.3) NOZ=1
DO 7914 IL=1,NOZ
JUL=IL
7914 CALL NNTOL(M,XX,IC,JUL,P,TI,RK)
CONTINUE
7913 CONTINUE
BRMJ=XX(I,KKK)
CALL ENTLIO(I+M,TI,M)
X1=RL(I-1);X2=RL(I);INDM=3
CALL REGF2(0.005,20,X1,X2,I,XX,VALUE,N,M,RL,V)
RLM(I)=VALUE
IF(KEY.NE.1) GO TO 4871
WRITE(22,4872) I,V(I+1),BRMJ,FACT2,VM(I)

```

```

4871 1. 8X,'FACT2=',E15.6,3X,'VM=',E15.6)
      CONTINUE
      DO 6020 IA=1,M
        SPRIME(1,I,IA)=(RK(I,IA)*(1.0+SV(I))*V(I))/RLM(I)
6020  CONTINUE
      IND=2
      X1=XX(I-1,KKK);X2=XX(I+1,KKK)
      DO 6030 IA=1,M
        XM(IA)=XX(I,IA)/RL(I)
6030  CONTINUE
      IB=1
      CALL REGF(X1,X2,0.0001,20,TCAL,IB)
      TI=TCAL
      IF(KEY.NE.0) GO TO 8879
      WRITE(22,8869) IND,TCAL
8869  FORMAT(2X,'IND=',I3,2X,'TCAL=',F10.5)
8879  CONTINUE
      IC=1
      CALL ENTLIO(IC,PI,M)
      IF(KTHERM.NE.1) GO TO 6912
      CALL EOLBM(TI,IC)
      GO TO 6913
6912  CONTINUE
      NOZ=3
      IF(KTHERM.EQ.3) NOZ=1
      DO 6914 JI=1,NOZ
        JIL=JI
        LINE=11305;TYPE*,LINE
        CALL NONIDL(M,XX,IC,JIL,P,TI,RK)
        LINE=11307;TYPE*,LINE
6914  CONTINUE
6913  CONTINUE
      CALL ENTIVAP(I+N+1,PI,M)
      X1=V(I);X2=V(I+1);INDM=4
      CALL REGF2(0.005,20,X1,X2,I,XX,VALUE,N,M,RL,V)
      VM(I)=VALUE
      DO 5040 IA=1,M
        SPRIME(2,I,IA)=RK(I,IA)*VM(I)/((1.0+SL(I))*RL(I))
      IF(KEY.NE.1) GO TO 5824
      WRITE(22,5724) I,IA,SPRIME(2,I,IA),VM(I),RK(I,IA)
5724  FORMAT(5X,'I=',I3,3X,'IA=',I3,3X,'SPRIME2=',E15.6/)
      1. 8X,'VM=',E15.6,4X,'RK(I,IA)=' ,E15.6)
5824  CONTINUE
      AA(1,IA)=SORT((SPRIME(1,I,IA)*(1.0+SPRIME(2,I,IA)))+0.25)-0.5
      TERM=(AA(1,IA))*((RMJ+1)
      DR(1,IA)=(AA(1,IA)-TERM)
      PIF(I,IA)=(1.0-TERM)
      ZX(1,IA)=(SPRIME(1,I,IA)*(SPRIME(2,I,IA)+1.0))/(SPRIME(1,I,IA)+
      1.0)
      IF (KEY.NE.1) GO TO 5040
      WRITE(22,8569) I,IA,SPRIME(1,I,IA),SPRIME(2,I,IA),DR(1,IA),
      1. PIE(1,IA)
8569  FORMAT(3X,'I=',I3,3X,'IA=',I3,3X,'SPRIME1=',E15.6,3X,'SPRIME2=',
      1. E15.6/8X,'DR=',E15.6,3X,'PIE=',E15.6)
6040  CONTINUE
5543  CONTINUE
      DO 100 I=1,N
        IB=1
        CALL G7NG(IB,KKK,N,XX)
        DO 100 J=1,KKK
          GP(I,J)=G(J)
100  CONTINUE
      RETURN
      END
      *****

```

```

*****
THOMAS :::
*****
SUBROUTINE THOMAS RETURNS THE VALUES OF UNKNOWN'S BY
THOMAS ALGORITHM FOR SIMPLE TRIDIAGONAL MATRICES
SUBROUTINE THOMAS(N,A,B,C,D,X)
DIMENSION A(25),B(25),C(25),D(25),X(25)
COMMON /AREA28/LOCK
P=(C(1)/B(1))
D=(D(1)/B(1))
B(1)=P
D(1)=0
DO 77 J=2,N
PR=(C(J)/(B(J)-B(J-1)*A(J)))
DR=(D(J)-A(J)*D(J-1))/(B(J)-B(J-1)*A(J))
B(J)=PR
D(J)=DR
CONTINUE
X(N)=D(N)
M=N-1
CONTINUE
X(M)=D(M)-(B(M)*X(M+1))
M=M-1
IF(M.GE.1) GO TO 50
RETURN
END
*****

```

00000

77

50

2

EOLBM:::

SUBROUTINE EOLBM RETURNS THE VALUES OF EOL. CONSTANTS
AND THEIR DERIVATIVE (W.R.T. TEM) FROM RELATIVE
VOLATILITIES DATA

SUBROUTINE EOLBM(I,J)

DIMENSION A1(20),A2(20),A3(20),RK(25,5),DERIV(25,5)

COMMON /AREA1/RK,M,P/AREA11/DERIV

COMMON /AREA28/LOCK/AREA33/NUMBER,KTHERM

DATA (A1(I),I=1,3)/.84,-.177,-.0879/

DATA (A2(I),I=1,3)/-46.6,49.5,17.7/

DATA (A3(I),I=1,3)/49.4,-4.15,-2031/

DATA (A4(I),I=1,3)/-3.033,2.22,1.31/

DO 10 I=1,M

RK(J,I)=A1(I)+(1.0E-04)*A2(I)*T+(1.0E-06)*A3(I)

T(1.0E-08)*A4(I)*(T**3.0)

IF(NUMBER.EQ.1) GO TO 10

DERIV(J,I)=(1.0E-04)*A2(I)+(2.0E-06)*A3(I)*T+(3.0E-08)*

A4(I)*(T**2)

CONTINUE

RETURN

END

ENTVAP:::

SUBROUTINE ENTVAP RETURNS THE VALUES OF VAP. ENTH. & HEAT
CAPACITIES

SUBROUTINE ENTVAP(I,T,M)

DIMENSION ENTV(25,5),E1(20),E2(20),E3(20),CPV(25,5)

COMMON /AREA2/ENTV/AREA9/CPV

COMMON /AREA28/LOCK/AREA33/NUMBER,KTHERM

DATA (E1(I),I=1,3)/1.195,1.655,1.920/

DATA (E2(I),I=1,3)/12.35,9.01,29.5/

DATA (E3(I),I=1,3)/2.21,3.78,.72/

IF(NUMBER.NE.1) GO TO 19

DO 10 I=1,M

ENTV(I,J)=E1(J)*(1.0E+04)+E2(J)*T+E3(J)*T*T*(1.0E-02)

ENTV(I,J)=ENTV(I,J)*(1.0E-04)

CONTINUE

GO TO 25

CONTINUE

DO 20 I=1,M

ENTV(I,J)=E1(J)*(1.0E+04)+E2(J)*T+E3(J)*T*T*(1.0E-02)

CPV(I,J)=E2(J)+E3(J)*(2.0E-02)*T

ENTV(I,J)=ENTV(I,J)*(1.0E-04)

CPV(I,J)=CPV(I,J)*(1.0E-04)

CONTINUE

CONTINUE

RETURN

END

ENTLQ:::

SUBROUTINE ENTLQ. RETURNS THE VALUES OF LQ. ENTH. &
HEAT CAPACITIES

SUBROUTINE ENTLQ(I,T,M)

DIMENSION ENTL(25,5),C1(20),C2(20),C3(20),CPL(25,5)

COMMON /AREA3/ENTL/AREA8/CPL

COMMON /AREA28/LOCK/AREA33/NUMBER,KTHERM

DATA (C1(I),I=1,3)/3.62,6.28,6.84/

DATA (C2(I),I=1,3)/40.5,25.0,38.5/

DATA (C3(I),I=1,3)/-1.32,4.65,2.52/

IF(NUMBER.NE.1) GO TO 19

DO 10 I=1,M

ENTL(I,J)=C1(J)*(1.0E+03)+C2(J)*T+C3(J)*(1.0E-02)*(T*T)

ENTL(I,J)=ENTL(I,J)*(1.0E-04)

CONTINUE

GO TO 25

CONTINUE

DO 20 I=1,M

ENTL(I,J)=C1(J)*(1.0E+03)+C2(J)*T+C3(J)*(1.0E-02)*(T*T)

CPL(I,J)=C2(J)+C3(J)*(2.0E-02)*T

ENTL(I,J)=ENTL(I,J)*(1.0E-04)

CPL(I,J)=CPL(I,J)*(1.0E-04)

CONTINUE

CONTINUE

RETURN

END

CHANGE ::::

SUBROUTINE CHANGE RETURNS THE DERIVATIVES FOR TEMP/VAP.
OR Q10. COM. FLOW RATES

SUBROUTINE CHANGE(M,I,N,XX)
DIMENSION XM(12),V(25),RL(25),FLC(25,5),FVC(25,5),DELTA(25,24)
1,XX(25,25),FL(25),FV(25)

COMMON /AREA7/FLC,FVC,REFLUX,DIS,RL,V,FL,FV/AREA15/XM/AREA23/
1, DELTA

COMMON /AREA17/IND

KEY=2

SETUP=0.0001

X1=XX(I-1,N)

X2=XX(I+1,N)

IF(IND.EQ.2) GO TO 100

DO 20 IA=1,N

XM(IA)=XX(I,IA+M)/V(I)

CONTINUE

CALL REGF(X1,X2,0.0001,20,TCAL,I)

SUMOLD=TCAL

DO 30 IB=M+1,N-1

XX(I,IB)=XX(I,IB)+SETUP

DO 40 IA=1,M

XM(IA)=XX(I,IA+M)/(V(I)+SETUP)

CONTINUE

CALL REGF(X1,X2,0.0001,20,TCAL,I)

DELTA(I,IB)=(TCAL-SUMOLD)/SETUP

XX(I,IB)=XX(I,IB)-SETUP

CONTINUE

GO TO 600

CONTINUE

DO 150 IA=1,M

XM(IA)=XX(I,IA)/RL(I)

CONTINUE

CALL REGF(X1,X2,0.0001,20,TCAL,I)

SUMOLD=TCAL

DO 160 IA=1,M

XX(I,IA)=XX(I,IA)+SETUP

DO 170 IT=1,M

XM(IT)=XX(I,IT)/(RL(I)+SETUP)

CONTINUE

CALL REGF(X1,X2,0.0001,20,TCAL,I)

DELTA(I,IA)=(TCAL-SUMOLD)/SETUP

XX(I,IA)=XX(I,IA)-SETUP

CONTINUE

CONTINUE

IF(KEY=3) 650,610,650

CONTINUE

DO 610 IA=1,N-1

WRITE(22,615) I,IA,DELTA(I,IA)

FORMAT(5X,'I=',I3,2X,'IA=',I3,2X,'DELTA(I,IA)=',E15.6)

CONTINUE

CONTINUE

RETURN

END

SUBROUTINE JACO3 ::

SUBROUTINE JACO3(XX,I,N,GAMMA)

DIMENSION FOLD(3),FNEW(5),DDR(25,5),DPAI(25,5),RK(25,5)

DIMENSION SP(2,20,5),SEF(20,5),DR(25,5),PIE(25,5),

1 GAMMA(25,5),XX(25,25),RL(25),V(25),FLC(25,5),

2 FVC(25,5),SPRIME(2,25,5),AA(25,5),FL(25),FV(25)

DIMENSION RLM(25),VM(25),RLDUM(25),RVM(25),ZX(25,5),ZZX(25,5)

COMMON /AREA7/FLC,FVC,REFLUX,DIS,RL,V,FL,FV

COMMON /AR20/DR/Y1/PIE,ZX/AREA21/AA,SPRIME/AREA1/RK,M,P

COMMON /AREA25/RLM,VM/AREA4/NO,IPRINT/AREA61/INDM

SETUP=0.005

KEY=2

RSTAGE=XX(J,N)+SETUP

RLDUM(J)=RLM(J)

RVM(J)=VM(J)

IF(KEY.NE.1) GO TO 2

WRITE(NJ,6666) RL(J),RL(J-1),V(J),V(J+1),RSTAGE,FACT11

FORMAT(5X,'RL J & J-1',2E15.6,'V J & J+1',2E15.6/

1, 3X,'RSTAGE=',F8.4,3X,'FACT11=',E15.6)

CONTINUE

IF(KEY.NE.1) GO TO 3

WRITE(22,2895),RLM(J),VM(J),RLDUM(J),RVM(J)

FORMAT(2X,4E19.3)

CONTINUE

DO 6020 IA=1,M

SP(1,J,IA)=(SPRIME(1,J,IA)*RLM(J)/RLDUM(J))

20

40

30

100

150

170

160

600

610

615

640

650

20

2

6666

2

2895

3

```

2387 IF(KEY.NE.1) GO TO 5
5 WRITE(22,2387) SP(1,J,IA),SP(2,J,IA),SPRIME(1,J,IA)
1, SPRIME(2,J,IA)
FORMAT(2X,4E19.3)
CONTINUE
SEF(J,IA)=SORT((SP(1,J,IA)*(1.0+SP(2,J,IA)))+0.25)-0.5
TERM=(SEF(J,IA))*(RSTAGE+1.0)
DDR(J,IA)=(SEF(J,IA)-TERM)
DPAI(J,IA)=(1.0-TERM)
ZZX(J,IA)=(SP(1,J,IA)*(SP(2,J,IA)+1.0))/(SP(1,J,IA)+1.0)
1087 IF(KEY.NE.1) GO TO 7
7 WRITE(22,1087) DR(J,IA),DDR(J,IA),TA,J
6020 FORMAT(2X,'DR=',E19.8,2X,'DDR=',E19.8,2X,'TA=',I3,2X,'J=',I3)
CONTINUE
CONTINUE
DO 6030 IA=1,M
L=IA+4
JJ=J-1
FOLD(IA)=((XX(JJ,IA)-XX(J,IA))/XX(JJ,IA))*PIE(J,IA)-
1 (1.0-((XX(J,IA)-XX(JJ,IA)+XX(J,L))/(XX(JJ,IA)*ZX(J,IA))))
2 *DR(J,IA)
FNEW(IA)=((XX(JJ,IA)-XX(J,IA))/XX(JJ,IA))*DPAI(J,IA)-
1 (1.0-((XX(J,IA)-XX(JJ,IA)+XX(J,L))/(XX(JJ,IA)*ZZX(J,IA))))
2 *DDR(J,IA)
GAMMA(J,IA)=(FNEW(IA)-FOLD(IA))/(SETUP)
GAMMA(J,IA)=GAMMA(J,IA)*XX(JJ,IA)
6025 IF(KEY.NE.1) GO TO 6030
FORMAT(5X,'J=',I3,3X,'IA=',I3,3X,'GAMMA(J,IA)=' ,E15.6)
6040 WRITE(ND,6040) IA,FOLD(IA),FNEW(IA)
6030 FORMAT(5X,'IA=',I3,3X,'FOLD=',E15.6,'FNEW=',E15.6)
CONTINUE
RETURN
END
*****
C
C SUBROUTINE BTYPE :: CALCULATES B-TYPE EQUILIBRIUM FUNCTION
C IN DESIGN PROBLEMS
SUBROUTINE BTYPE(ZX,DR,PIE,XX,M,J,RV,FCAL)
DIMENSION XX(25,25),DR(25,5),PIE(25,5),FCAL(5)
DIMENSION RV(20),ZX(25,5)
DO 10 IA=1,M
JJ=J-1
L=IA+4
FCAL(IA)=((XX(JJ,IA)-XX(J,IA))/XX(JJ,IA))*PIE(J,IA)-
1 (1.0-((XX(J,IA)-XX(JJ,IA)+XX(J,L))/(XX(JJ,IA)*ZX(J,IA))))
2 *DR(J,IA)
FCAL(IA)=FCAL(IA)*XX(JJ,IA)
10 CONTINUE
RETURN
END
*****
C
C SUBROUTINE JACOB2 :::::
SUBROUTINE JACOB2(XX,J,KKK,ML,IAD,INDEX,KJ,ALPHA)
DIMENSION FOLD(5),FNEW(5),RV(25),ALPHA(12,12),RLM(25),VM(25)
DIMENSION DRDM(25),RVM(25),SP(2,20,5),DDR(25,5),DAA(25,5)
1 DIMENSION XX(25,25),GT(25,25),RK(25,5),FLC(25,5),FVC(25,5),
2 RL(25),V(25),FL(25),FV(25),A(25,25),B(25,25),C(25,150),D(25,60)
3 G(25),ENTV(25,5),ENTL(25,5)
DIMENSION DR(25,5),PIE(25,5),XM(5),DPIE(25,5)
DIMENSION SPRIME(2,25,5),SL(20),SV(20),EFI(25,5)
DIMENSION RRL(25),ZX(25,5),ZZX(25,5),CPV(25,5),CPL(25,5)
DIMENSION TCP(5)
DIMENSION AA(25,5),IMP(5),TTL(5),TTV(5),TCP(5),TCPV(5)
COMMON /AREA15/KM/AREA17/IND/AR20/DR/Y1/PIE,ZX/AREA21/AA,SPRIME
COMMON /AREA1/RK,M,P/AREA7/FLC,FVC,REFLUX,DIS,RL,V,FL,FV/
1 AREAS/A,B,C,D,G/AREA2/ENTV/AREA3/ENTL/AREA8/CPL/AREA9/CPV
COMMON /AREA10/EFI,SL,SV/AREA14/ITN/AREA25/RLM,VM
COMMON /AREA28/LOCK/AREA29/NUMBER/AREA61/INDM
COMMON /AREA33/NUMBER,KTHERM/AREA80/NJ
KEY=2
SETUP=0.1
RV(J)=V(J)
CALL BTYPE(ZX,DR,PIE,XX,M,J,RV,FOLD)
DO 5841 IA=1,M
PIL(IA)=ENTL(J-1,IA)
PTV(IA)=ENTV(J+1,IA)
TCP(IA)=CPL(J-1,IA)
TCPV(IA)=CPV(J+1,IA)
5841 CONTINUE
XX(KJ,IAD)=XX(KJ,IAD)+SETUP
RV(J+1)=V(J+1)
RRL(J-1)=RL(J-1)
RRL(J)=RL(J)
IF(INDEX.EQ.3) CALL ENTLIO(J-1,XX(KJ,IAD),M)
IF(INDEX.EQ.4) CALL ENTVAP(J+1,XX(KJ,IAD),M)
IF(INDEX.EQ.1) RRL(KJ)=RL(KJ)+SETUP
IF(INDEX.EQ.2) RV(KJ)=V(KJ)+SETUP

```

```

1. 3X, 'VM=', E15.6, 4X, 'RK(I,IA)=' , E15.6)
5824 CONTINUE
DAA(J,IA)=SORT((SP(1,J,IA)*(1.0+SP(2,J,IA)))+0.25)-0.5
TERM=(DAA(J,IA))* (RRMJ+1.0)
DDR(J,IA)=(DAA(J,IA)-TERM)
DPIE(J,IA)=(1.0-TERM)
ZZX(J,IA)=(SP(1,J,IA)*(SP(2,J,IA)+1.0))/(SP(1,J,IA)+1.0)
IF (KEY.NE.1) GO TO 6040
WRITE(22,8569) J,IA,SP(1,J,IA),SP(2,J,IA),DDR(J,IA),
1. DPIE(J,IA)
8569 FORMAT(3X, 'I=', I3, 3X, 'IA=', I3, 3X, 'SPRIME1=', E15.6, 3X, 'SPRIME2=',
1. E15.6/8X, 'DR=', E15.6, 3X, 'PIE=', E15.6)
6040 CONTINUE
CALL CBTYP(ZZX,DDR,DPIE,XX,M,J,RV,FNEW)
DO 5848 IA=1,M
ENTL(J-1,IA)=ITL(IA)
ENTV(J+1,IA)=ITV(IA)
TCL(J-1,IA)=TCPL(IA)
CPV(J+1,IA)=TCPV(IA)
5848 CONTINUE
XX(KJ,IA)=XX(KJ,IA)-SETUP
DO 2343 NN=1,ML
ALPHA(NN,IA)=(FNEW(NN)-FOLD(NN))/(SETUP)
2343 CONTINUE
RETURN
END
*****
C

```

CCCCC

```

*****
BTHOM::: \
*****
SUBROUTINE BTHOM RETURNS THE VALUES OF UNKNOWN BY
BLOCK TRIAGONAL THOMAS ALGORITHM
"C" MATRIX==>DEL(X)
SUBROUTINE BTHOM(X,N,NJ)
      DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25)
1      X(25,25),Y(25,25),XX(25,25)
      DIMENSION DELTA(25,24)
      COMMON /AREA4/NO,IPRINT
      COMMON /AREA5/A,B,C,D,G/AREA6/X,Y/AREA14/ITN/AREA12/SUMMM
      COMMON /AREA23/DELTA/AREA17/IND
      COMMON /AREA28/LOCK
      M=(N-1)/2
      DO 55 J=1,NJ
      IF(LOCK.EQ.2) GO TO 545
      IF((J.EQ.3).OR.(J.EQ.9)) GO TO 540
545      CONTINUE
      GO TO 55
540      CONTINUE
      DO 550 IND=1,2
      JB=J
      CALL CHANGE(M,JB,N,XX)
550      CONTINUE
55      CONTINUE
      DO 5 J=1,NJ
      JB=J
      IF(ITN.EQ.1) CALL GENG(JB,N,NJ,XX)
      IF(ITN.NE.1) CALL GENG(JB,N)
      DO 20 IA=1,N
      SUMMM=SUMMM+ABS(G(IA))
20      CONTINUE
      CALL GENA(JB,N,NJ,XX)
      CALL GENB(JB,N,NJ,XX)
      CALL GENC(JB,N,NJ,XX)
      IF(J.EQ.1) CALL GENX(JB,N)
      IF(J.EQ.NJ) CALL GENY(JB,N)
      IF(ITN.NE.1) GO TO 100
90      WRITE(22,90) J
      FORMAT(2X,'STAGE NO.=' ,I3)
      DO 120 IB=1,N
      WRITE(22,110) G(IA),IA
110      FORMAT(2X,'G=' ,E11.5,2X,'IA=' ,I3)
      DO 130 IB=1,N
      IF(A(IA,IB).EQ.0.0) GO TO 140
      WRITE(22,150) A(IA,IB),IA,IB
150      FORMAT(2X,'A=' ,E11.5,2X,'IA=' ,I3,2X,'IB=' ,I3)
140      CONTINUE
      IF(B(IA,IB).EQ.0.0) GO TO 160
      WRITE(22,170) B(IA,IB),IA,IB
170      FORMAT(2X,'B=' ,E11.5,2X,'IA=' ,I3,2X,'IB=' ,I3)
160      CONTINUE
      IF(C(IA,IB).EQ.0.0) GO TO 180
      WRITE(22,190) C(IA,IB),IA,IB
190      FORMAT(2X,'C=' ,E11.5,2X,'IA=' ,I3,2X,'IB=' ,I3)
180      CONTINUE
130      CONTINUE
120      CONTINUE
100      CONTINUE
      JB=J
      CALL BAND(JB,N,NJ,XX)
5      CONTINUE
      WRITE(25,5245)
5245      FORMAT(15X,'BLOCK THOMAS ENDS _____')
      RETURN
      END
*****

```

CCCCC

```

BAND:::
*****
SUBROUTINE BAND
SUBROUTINE BAND(J,N,NJ,XX)
      DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25),
1      X(25,25),Y(25,25),XX(25,25)
      DIMENSION E(21,22,40)
      COMMON /AREA5/A,B,C,D,G/AREA4/NO,IPRINT/AREA6/X,Y
      COMMON /AREA28/LOCK
101      FORMAT(15X,DETERRN=J AT J= ,I4)
      ITT=2*N+1
      IF(J-2) 1,6,8
1      NP1=N+1
      DO 2 I=1,N
      DO(I,ITT)=G(I)
      DO 2 L=1,N
      LON=L+N
      D(I,LON)=X(I,L)
2      CONTINUE
      RETURN

```

2

```

1114  FORMAT(5X,'LINE:3450')
      CALL MATINV(N,ITT,DETERM)
      WRITE(05,1115)
1115  FORMAT(5X,'LINE:3500')
      IF(DETERM) 4,3,4
3      WRITE(05,101) J
4      DO 5 K=1,N
        E(K,NP1,1)=D(K,ITT)
        DO 5 L=1,N
          E(K,L,1)=-D(K,L)
          LPN=LN+N
          X(K,L)=-D(K,LPN)
5      CONTINUE
      RETURN
6      DO 7 I=1,N
        DO 7 K=1,N
        DO 7 L=1,N
          D(I,K)=D(I,K)+A(I,L)*X(L,K)
7      CONTINUE
8      IF(J-NJ) 11,9,9
9      DO 10 I=1,N
        DO 10 L=1,N
          G(I)=G(I)-Y(I,L)*E(L,NP1,J-2)
        DO 10 M=1,N
          A(I,L)=A(I,L)+Y(I,M)*E(M,L,J-2)
10     CONTINUE
11     DO 12 I=1,N
        D(I,NP1)=-G(I)
        DO 12 L=1,N
          D(I,NP1)=D(I,NP1)+A(I,L)*E(L,NP1,J-1)
        DO 12 K=1,N
          B(I,K)=B(I,K)+A(I,L)*E(L,K,J-1)
12     CONTINUE
      WRITE(05,1116)
1116  FORMAT(5X,'LINE:6050')
      CALL MATINV(N,NP1,DETERM)
      WRITE(05,1117)
1117  FORMAT(5X,'LINE:6100')
      IF(DETERM) 14,13,14
13     WRITE(05,101) J
14     DO 15 K=1,N
        DO 15 M=1,NP1
          E(K,M,J)=-D(K,M)
15     CONTINUE
      IF(J-NJ) 20,16,16
16     DO 17 K=1,N
        C(K,J)=E(K,NP1,J)
17     CONTINUE
        DO 18 J=2,NJ
          M=NJ-J+1
          DO 18 K=1,N
            E(K,M)=E(K,NP1,M)
            DO 18 L=1,N
              C(K,M)=C(K,M)+E(K,L,M)*C(L,M+1)
18     CONTINUE
        DO 19 L=1,M
          DO 19 K=1,N
            C(K,J)=C(K,1)+X(K,L)*C(L,3)
19     CONTINUE
20     RETURN
      END
      *****
      MATINV:::
      *****
      SUBROUTINE FOR MATRIX INVERSION
      SUBROUTINE MATINV(N,M,DETERM)
      DIMENSION A(25,25),B(25,25),C(25,150),D(25,60),G(25),
1      X(25,25),Y(25,25),XX(25,25)
      DIMENSION ID(25)
      COMMON /AREA5/A,B,C,D,G/AREA4/NO,IPRINT/AREA6/X,Y
      COMMON /AREA28/LOCK

      DETERM=1.0
      DO 1 I=1,N
        ID(I)=0
1      CONTINUE
      DO 18 NN=1,N
        BMAX=1.0
        DO 5 I=1,N
          IF(ID(I).NE.0) GO TO 6
          BNEXT=0.0
          BTRY=0.0
          DO 5 J=1,N
            IF(ID(J).NE.0) GO TO 5
            IF(ABS(B(I,J)).LE.BNEXT) GO TO 5
            BNEXT=(ABS(B(I,J)))
            IF(BNEXT.LE.BTRY) GO TO 5
            BTRY=BTRY
            BTRY=ABS(B(I,J))

```

```

5      JC=J
      CONTINUE
      IF(BNEXT.GE.BMAX*BTRY) GO TO 6
      BMAX=BNEXT/BTRY
      IRJW=I
      JCOL=JC
6      CONTINUE
      IF(1D(JC).EQ.0) GO TO 8
      DEFORM=0.0
      RETURN
8      CONTINUE
      ID(JCOL)=1
      IF(JCOL.EQ.IRJW) GO TO 12
      DO 10 J=1,N
      SAVE=B(IRJW,J)
      B(IRJW,J)=B(JCOL,J)
      B(JCOL,J)=SAVE
10     CONTINUE
      DO 11 K=1,M
      SAVE=D(IRJW,K)
      D(IRJW,K)=D(JCOL,K)
      D(JCOL,K)=SAVE
11     CONTINUE
12     CONTINUE
      F=1.0/B(JCOL,JCOL)
      DO 13 J=1,N
      B(JCOL,J)=B(JCOL,J)*F
13     CONTINUE
      DO 14 K=1,M
      D(JCOL,K)=D(JCOL,K)*F
14     CONTINUE
      DO 15 I=1,N
      IF(1.EQ.JCOL) GO TO 18
      F=B(I,JCOL)
      DO 16 J=1,N
      B(I,J)=B(I,J)-F*B(JCOL,J)
16     CONTINUE
      DO 17 K=1,M
      D(I,K)=D(I,K)-F*D(JCOL,K)
17     CONTINUE
18     CONTINUE
      RETURN
      END
      *****

```

```

*****
SUBROUTINE TO CALCULATE NUMERICAL DERIVATIVE OF EQUILIBRIUM
CONSTANT W.R.T TEMPERATURE OR MOLE FRACTIONS  ::::
-----

```

```

SUBROUTINE NUME3B(T,J,XX)
DIMENSION DERIV(25,5),RK(25,5),DUMRK(2,25,5),XX(25,25)
DIMENSION XETAX(20,3,3),XETAY(20,3,3)
COMMON /AREA1/RK,M,P/AREA11/DERIV/AREA28/LOCK
COMMON /AREA33/NUMBER,KTHERM/AREA37/XETAX,XETAY
SETUP=0.0001
DT=T
DO 10 KK=1,2
IF(KTHERM.EQ.2) GO TO 200
CALL EQLBM(DT,J)
DO 10 12
CONTINUE
CALL NONIDC(M,XX,J,1,P,DT,RK)
CONTINUE
DO 5 II=1,M
DUMRK(KK,J,II)=RK(J,II)
CONTINUE
DT=DT+SETUP
CONTINUE
DO 20 II=1,M
DERIV(J,II)=(DUMRK(2,J,II)-DUMRK(1,J,II))/SETUP
CONTINUE
IF(KTHERM.EQ.1) GO TO 500
DO 33 MB=1,M
DO 32 NN=1,2
CALL NONIDC(M,XX,J,1,P,T,RK)
DO 31 II=1,M
DUMRK(NN,J,II)=RK(J,II)
CONTINUE
XX(J,MB)=XX(J,MB)+SETUP
CONTINUE
XX(J,MB)=XX(J,MB)-(SETUP*2)
DO 34 NK=1,M
XETAX(J,NK,MB)=(DUMRK(2,J,NK)-DUMRK(1,J,NK))/SETUP
CONTINUE
CONTINUE
DO 43 MB=M+1,2*M
DO 42 NN=1,2
CALL NONIDC(M,XX,J,1,P,T,RK)
DO 41 II=1,M
DUMRK(NN,J,II)=RK(J,II)
CONTINUE
XX(J,MB)=XX(J,MB)+SETUP
CONTINUE
XX(J,MB)=XX(J,MB)-(SETUP*2)
DO 44 NK=1,M
XETAY(J,NK,MB-M)=(DUMRK(2,J,NK)-DUMRK(1,J,NK))/SETUP
CONTINUE
CONTINUE
CONTINUE
RETURN
END

```

```

*****
SUBROUTINE TO CALCULATE NUMERICAL DERIVATIVE OF ENTHALPY (LIQUID
& VAPOR) W.R.T TEMPERATURE  ::::
-----

```

```

SUBROUTINE NUMENT(T,J,M)
DIMENSION ENTV(25,5),CPV(25,5)
DIMENSION ENTL(25,5),CPL(25,5)
DIMENSION DUMVAP(2,25,5),DUMLIO(2,25,5)
COMMON /AREA3/ENTL/AREA8/CPV
COMMON /AREA2/ENTV/AREA9/CPV
COMMON /AREA23/LOCK/AREA33/NUMBER,KTHERM
SETUP=0.0001
DT=T
IF(KTHERM.EQ.2) GO TO 25
DO 10 KK=1,2
IK=J
CALL ENTVAP(IK,DT,M)
CALL ENTLIO(IK,DT,M)
DO 5 II=1,M
DUMVAP(KK,J,II)=ENTV(J,II)
DUMLIO(KK,J,II)=ENTL(J,II)
CONTINUE
DT=DT+SETUP
CONTINUE
IF(KTHERM.EQ.2) GO TO 25
DO 20 II=1,M
CPV(J,II)=(DUMVAP(2,J,II)-DUMVAP(1,J,II))/SETUP
CPL(J,II)=(DUMLIO(2,J,II)-DUMLIO(1,J,II))/SETUP
CONTINUE
CONTINUE

```

 SUBROUTINE FOR THERMODYNAMIC PROPERTY ESTIMATION ::::

SUBROUTINE NONIDL(NV,XX,JSP,INDEX,PTZ,TZ,RK)
 DIMENSION PC(5),TC(5),ZC(5),OMEGA(5),VC(5),X(5),Y(5),KI(5)
 DIMENSION CPVAPA(5),CPVAPB(5),CPVAPC(5),CPVAPD(5)
 DIMENSION DELHV(10),YDUM(10)
 DIMENSION ENTL(25,5),ENTV(25,5),CPL(25,5),CPV(25,5)
 DIMENSION RK(25,5),XX(25,25)
 COMMON /MADAT/N,NCOMP,KSITU,NR,NPRI
 COMMON /SYSDAT/PC,TC,ZC,OMEGA,VC
 COMMON /AREA2/ENTV/AREA9/CPV/AREA3/ENTL/AREA8/CPL

REAL KI
 DATA KEY/2/
 DATA NCMP,KSITU,NR,NPRI/3,3,0,0/
 ----- IDEAL GAS HEAT CAPACITY EQUATION CONSTANTS :::::
 DATA (CPVAPA(K),K=1,3)/-1.009,2.266,-0.866/
 DATA (CPVAPB(K),K=1,3)/7.315E-02,7.913E-02,1.164E-01/
 DATA (CPVAPC(K),K=1,3)/-3.789E-05,-2.647E-05,-6.613E-05/
 DATA (CPVAPD(K),K=1,3)/7.978E-09,-0.674E-09,1.267E-08/
 ----- MOLECULAR DATA FOR NONIDEAL THERMODYNAMICS :::::
 DATA (PC(K),K=1,3)/41.9,37.5,33.3/
 DATA (TC(K),K=1,3)/369.8,425.2,469.6/
 DATA (ZC(K),K=1,3)/0.281,0.274,0.262/
 DATA (OMEGA(K),K=1,3)/0.152,0.274,0.251/
 ----- IF INDEX=2 ; VAPOUR ENTHALPY IS CALCULATED
 INDEX=1 ; EQUILIBRIUM CALCULATION IS DONE
 INDEX=3 ; LIQUID ENTHALPY IS ONLY CALCULATED

N=NV
 PT=PTZ/14.7
 T=((TZ-32.0)*(5.0/9.0))+273.15
 IF(INDEX.GE.2) GO TO 275
 IF(JSP.EQ.1) RETURN
 SUM1=0.0
 SUM2=0.0
 DO 126 K=1,N
 SUM1=SUM1+XX(JSP,K)
 CONTINUE
 RLIO=SUM1
 DO 127 K=N+1,2*N
 SUM2=SUM2+XX(JSP,K)
 CONTINUE
 VAP=SUM2
 DO 128 K=1,N
 X(K)=XX(JSP,K)/RLIO
 Y(K)=XX(JSP,K+N)/VAP
 CONTINUE
 CALL EQUIL(T,PT,X,Y,KI)
 DO 129 K=1,N
 RK(JSP,K)=KI(K)
 CONTINUE
 IF(NPRI.EQ.0) GO TO 500
 IF(KSITU.EQ.1) GO TO 20
 WRITE(22,10)
 FORMAT(15X,'EQLBM CONST BY PENG-ROBINSON EQN.')

GO TO 30
 WRITE(22,25)
 FORMAT(15X,'EQLBM CONST BY SOAVE-REDLICH-KWONG EQN.')

CONTINUE
 DO 40 NN=1,N
 WRITE(22,35) JSP,KI(NN),NN
 FORMAT(5X,'JSP=',I3,2X,'KI(N)=',E16.8,4X,'FOR COMP. NO. = ',I3)
 CONTINUE
 GO TO 500
 CONTINUE
 IF(INDEX.EQ.3) GO TO 278
 IF(JSP.EQ.1) RETURN
 DO 300 I=1,N
 DO 276 LL=1,N
 YDUM(LL)=0.0
 IF(I.EQ.LL) YDUM(I)=1.0
 CONTINUE
 CALL ZOOT(ZMAX,ZMIN,YDUM,F,B,FM,TC,PC,T,PT,OMEGA,Z,N,KSITU)
 PCDAT=PC(I)
 TCDAT=TC(I)
 ZCDAT=ZC(I)
 CPA=CPVAPA(I)
 CPB=CPVAPB(I)
 CPC=CPVAPC(I)
 CPD=CPVAPD(I)
 CALL ENTHLP(PT,T,Z,TCDAT,PCDAT,ZCDAT,CPA,CPB,CPC,CPD,HACT,DELH,
 1 CPVDAT)
 CPV(JSP,I)=CPVDAT
 ENTV(JSP,I)=HACT
 DELHV(I)=DELH
 IF(KEY.EQ.0) GO TO 300
 WRITE(22,280) JSP,I,ENTV(JSP,I),DELHV(I),CPV(JSP,I)
 FORMAT(15X,'ENTV(JSP,I)=',E15.8,3X,'DELHV=',E15.8,

1 3X, 'CPV=',E15.8)

CONTINUE

GO TO 500

CONTINUE

----- LIQUID ENTHALPY CALCULATION STARTS USING
EQUATION OF CORRESPONDING STATES :::

DO 310 I=1,N

OMDAT=OMEGA(I)

TCDAT=TC(I)

PCDAT=PC(I)

CPA=CPVAPA(I)

CPB=CPVAPB(I)

CPC=CPVAPC(I)

CPD=CPVAPD(I)

CALL ENTL2(TCDAT,PCDAT,T,PT,CPA,CPB,CPC,CPD,OMDAT,CPLDAT,DELHL,
1 DELHVT)

ENTL(JSP,I)=DELHL

CPL(JSP,I)=CPLDAT

IF(KEY.EQ.0) GO TO 310

WRITE(22,293) JSP,I,ENTL(JSP,I),DELHVT,CPL(JSP,I)

FORMAT(5X,I4,3X,'COMP.NO=',13,3X,'HL=',E15.8,3X,'DELHVT=',E15.8,
1 3X,'CPL=',E15.8)

CONTINUE

CONTINUE

RETURN

END

SUBROUTINE EQUIL(T,PT,X,Y,KI)

VARIABLES:

T:TEMP IN K

PT:PRESSURE IN ATM

X,Y:MOLE FRACTION IN LIQUID & VAPOR PHASE

KI:EQUILIBRIUM RATIOS

THIS SUBROUTINE DETERMINES WHETHER TO USE UNIFAC,SRK,PR FOR
FINDING
EQUILIBRIUM RATIOS AND THEN CALLS THE APPROPRIATE SUBROUTINE

SUBROUTINE EQUIL(T,PT,XXX,YYY,KII)

DIMENSION X(5),Y(5),KI(5),XXX(5),YYY(5),KII(5)

COMMON/MADAT/N,NCOMP,KSITU,NR,NPRI

REAL KI,KII

DO 1000 I=1,N

X(I)=XXX(I)

Y(I)=YYY(I)

GO TO(100,9,100)KSITU

SRK AND PR METHODS

IF (NPRI.EQ.1) GO TO 11

IF (NR.EQ.1) GO TO 11

WRITE(22,5)

FORMAT(5X,65(1H*)/,5X,1H*,63X,1H*/,5X,1H*,63X,1H*)

IF(KSITU.EQ.3) GO TO 113

WRITE(22,3)

FORMAT(5X,1H*,8X,'RKS-EQNS USED FOR CALCULATING EQUILIBRIUM RAT',
1 10S',7X,1H*)

GO TO 117

WRITE(22,114)

FORMAT(5X,1H*,8X,'PENG ROBINSON USED FOR CALCULATING EOILIBR
1 IUM RATIOS',9X,1H*)

WRITE(22,7)

NR=NR+1

CALL EQUIL2(X,Y,T,PT,KII,N,KSITU)

GO TO 10

CONTINUE

UNIFAC TO BE USED

WRITE(22,4)

FORMAT(5X,1H*,8X,'UNIFAC USED FOR CALCULATING EQUILIBRIUM RATIOS',
1 9X,1H*)

FORMAT(5X,1H*,63X,1H*/,5X,1H*,63X,1H*/,5X,65(1H*)//)

CONTINUE

RETURN

END

C*****
C*****

SUBROUTINE EQUIL2(N,NCOMP,T,PT,X,Y)

VARIABLES:

K1: INTERACTION PARAMETERS
 PHIV: FUGACITY OF THE VAPOR PHASE
 PHIL: FUGACITY OF THE LIQUID PHASE
 K: EQUILIBRIUM RATIOS
 Z: COMPRESSIBILITY

THIS PROGRAM CALCULATES THE EQUILIBRIUM RATIOS USING RKS EON'S
 OR PR EQNS DEPENDING UPON KSITU=10R3

SUBROUTINE EQUIL2(X,Y,T,PT,K,N,KSITU)

DIMENSION X(5),Y(5),K1(5,5)
 DIMENSION A(5),K(5),PHIV(5),PHIL(5),XXI(5),YYI(5)
 REAL K1,K
 COMMON/SYSDAT/PC(5),TC(5),ZC(5),OMEGA(5),VC(5)
 COMMON/EQIDAT/F(5),B(5),FM,BM,AA(5),AM,ASTAR,BSTAR

K1(I,J): INTERACTION PARAMETERS

DO 911 I=1,N
 DO 911 J=1,N
 K1(I,J)=0.0
 CONTINUE

STORING X'S&Y'S

DO 623 I=1,N
 XXI(I)=X(I)
 YYI(I)=Y(I)
 CONTINUE

NO=0: WHEN THIS SUBROUTINE IS EXECUTED FOR VAPOR
 =1: WHEN THIS SUBROUTINE IS EXECUTED FOR LIQUID

NO=0

R=82.05

SUBROUTINE ROOT CALLED ONCE FOR LIQ. AND ONCE FOR VAPOR

CALL ROOT(ZMAX,ZMIN,K1,Y,T,PT,Z,N,NO,KSITU)

WRITE(22,1431) ZMAX,ZMIN

FORMAT(5X,'ZMAX=',F7.4,'ZMIN=',F7.4)

IF ((ZMAX.LT.0.0)) GOTO 9999

IF (NO.EQ.1) GOTO 12

FOR LIQUID PHASE ZMIN CORRESPONDS TO ZMAX

ZMAX=ZMIN

V: VOLUME

V=((ZMAX)*R*T/PT)

GO TO (2000,1000,2002)KSITU

GA=0.4274803

GB=0.0866403

DO 2 I=1,N

A(I)=(GA*R*T*B(I)*F(I)/GB)

AM=(GA*R*T*BM*FM/GB)

DO 3 I=1,N

BABA=0.0

DO 4 J=1,N

GO 10(3000,4,3001)KSITU

BABA=BABA+((1.0-K1(I,J))*((A(I)*A(J))**0.5)*Y(J)/AM)

GO TO 4

BABA=BABA+2.*(1.-K1(J,I))*SQRT(AA(J)*AA(I))*Y(J)/AM

CONTINUE

PHILN: LIQUID FUGACITY

GO TO (3010,1000,3020)KSITU

PHILN=(B(I)*(ZMAX-1.0)/BM)+ALOG(V/(ZMAX*(V-BM)))+(AM/(BM*R*T))

1*ALOG((V+BM)/V)*((B(I)/BM)-2.0*BABA)

GO TO 3021

PHILN=B(I)*(ZMAX-1.0)/BM-ALOG(ZMAX-BSTAR)-ASTAR/(2.828*BSTAR)

1*(BABA-B(I)/BM)*ALOG((ZMAX+2.414*BSTAR)/(ZMAX-0.414*BSTAR))

CONTINUE

IF (NO.EQ.1) GOTO 5

DEPENDING ON NO: ASSIGN THE FUGACITY

PHIV(I)=EXP(PHILN)

GO TO 3

CONTINUE

PHIL(I)=EXP(PHILN)

CONTINUE

REPEAT THE LOOP FOR THE LIQUID

DO 6 I=1,N

Y(I)=X(I)

CONTINUE

IF (NO.EQ.2) GOTO 10

GOTO 1

```

10 CONTINUE
   CALCULATION OF BOTH LIQUID AND VAPOR FUGACITY COMPLETE
   REGENERATE X,Y
   DO 18 I=1,N
   X(I)=XXI(I)
   Y(I)=YYI(I)
18 CONTINUE
   DO 11 I=1,N
   WRITE(22,1731) I, PHIL(I),I,PHIV(I)
1731 FORMAT(5X,'PHILIO(',I2,')=',E16.9,'PHIVAP(',I2,')=',F7.5/)
   K(I)=(PHIL(I)/PHIV(I))
   WRITE(22,102),I,K(I)
102 FORMAT(' K(',I2,')=',F15.9)
11 CONTINUE
   GOTO 1000
   IF THE REAL ROOT IN THE CASE OF ONE REAL IS NEGATIVE -PRINT
   ERROR &STOP
9999 WRITE(22,55)
55 FORMAT(' ERROR!@#$$%^& CHECK?? CHECK ????? ')
   STOP
1000 CONTINUE
   RETURN
   END
*****
      SUBROUTINE ROOT(ZMAX,ZMIN,K1,Y,T,P,Z,N,NO,KSITU)
*****
      VARIABLES:
      TR:REDUCED TEMPERATURE
      PR:REDUCED PRESSURE
*****
      THIS SUBROUTINE CALCULATES THE COMPRESSIBILITY FACTOR,Z BY
      EVALUATING THE ROOTS OF THE CUBIC EQUATION AND THEN ASSIGNING
      THE LARGEST REAL ROOT TO THE VAPOR PHASE AND THE SMALLEST
      ROOT TO THE LIQUID PHASE

      SUBROUTINE ROOT(ZMAX,ZMIN,K1,Y,T,P,Z,N,NO,KSITU)
      DIMENSION Y(5),TR(5),PR(5)
      DIMENSION TEMP(5),K1(5,5)

      COMMON/SYSDAT/PC(5),TC(5),ZC(5),OMEGA(5),VC(5)
      COMMON/EQIDAT/F(5),B(5),FM,BM,AA(5),AM,ASTAR,BSTAR

      REAL K1
      R:GAS CONSTANTS
      P=82.05
      IF(KSITU.EQ.3) GO TO 2000
      SRK EQHS USED, KSITU=1
      GA,GB:CONSTANTS IN RKS
      GB=0.0866403
      GA=0.4274803
      RM=0.0
      FM1=0.0
      CALCULATE REDUCED TEMPERATURE & PRESSURE
      DEN=0.0
      DO 1 I=1,N
      TR(I)=T/TC(I)
      PR(I)=P/PC(I)
      CALCULATE DIFFERENT VARIABLES OF RKS
      B(I)=GB*R*TC(I)/(PC(I))
      BM=BM+Y(I)*B(I)
      TEMP(I)=(1.0+(0.480+1.574*OMEGA(I)-0.176*OMEGA(I)*OMEGA(I))*(1.0
      1-((TR(I))*0.5)))
      F(I)=(1.0/TR(I))*((TEMP(I))**2.0)
      CONTINUE
      CALCULATION OF VARIABLES (CONT.)
      DO 2 I=1,N
      DO 3 J=1,N
      FM1=Y(I)*Y(J)*(((TC(I)*F(I)*TC(J)*F(J))/(PC(I)*PC(J)))*0.5)*(1.
      10-K1(I,J))+FM1
      CONTINUE
      DEN=DEN+((Y(I)*TC(I))/PC(I))
      CONTINUE
      CALCULATION OF VARIABLES (CONT.)
      FM=(FM1/DEN)
      BSTAR=(BM*P)/(R*T)
      ASTAR=((BSTAR*GA)/GB)*FM
      CALCULATING ROOTS OF THE CUBIC EQUATION
      B1=-1.0
      C1=(ASTAR-BSTAR-(BSTAR*BSTAR))
      D1=-(ASTAR*BSTAR)
      GO TO 2001
      CONTINUE
      BM=0.
      DO 2002 I=1,N
      TR(I)=T/TC(I)

```

```

B(I)=0.07780*R*TC(I)/PC(I)
BM=BM+B(I)*Y(I)
AA(I)=(0.45724*((R*TC(I))**2)/PC(I))
AA(I)=(1.+(0.37464+1.54226*OMEGA(I)-0.26992*OMEGA(I)*OMEGA(I)
1)*(1.-((TR(I))*0.5))**2)*AA(I)
2002 CONTINUE
AM=0
DO 2003 I=1,N
DO 2003 J=1,N
2003 AM=AM+Y(I)*Y(J)*(1.-K1(I,J))*SORT(AA(I)*AA(J))
ASTAR=AM*P/(R*T*R*T)
BSTAR=BM*P/(R*T)
B1=BSTAR-1.0
C1=ASTAR-3.0*(BSTAR*BSTAR)-2.0*BSTAR
D1=-BSTAR*(ASTAR-BSTAR-(BSTAR*BSTAR))
CONTINUE
2001 PP=(3.0*C1-(B1*B1))/3.0
QQ=(27.0*D1-9.0*B1*C1+2.0*B1*B1*B1)/27.0
RR=(PP*PP*PP)/27.0+QQ*QQ/4.0
C RR: DETERMINES WHICH PATH TO TAKE
IF ((ABS(RR)).GT.0.1E-20) GOTO 155
GO TO 20
155 IF (RR) 10,20,30
10 CONTINUE
C WRITE(22,95)
C R<0
95 FORMAT(' THERE ARE THREE UNEQUAL REAL ROOTS OF THE CUBIC EQTN. I
1 N Z')
XS=((QQ*QQ/4.0)/((-PP*PP*PP)/27.0))
C XS=SQRT(XS)
THIS GIVES IN RADIANS
PHI=ACOS(XS)
AMP=2.0*((-PP/3.0)**0.5)
C ANG=(PHI/3.0)
CALCULATE ROOTS WHEN THREE REAL UNEQUAL ROOTS
X1=AMP*(COS(ANG))
X2=AMP*(COS(ANG+2.094395))
X3=AMP*(COS(ANG+4.18879))
IF (QQ.LT.0.0) GOTO 111
X1=-X1
X2=-X2
X3=-X3
111 X1=X1-(B1/3.0)
X2=X2-(B1/3.0)
X3=X3-(B1/3.0)
C WRITE(22,12) X1,X2,X3
12 FORMAT(' PDOTS ARE:',10X,'X1=',F10.6,3X,'X2=',F10.6,3X,'X3=',F10
1.6)
C FIND THE MAXIMUM & MINIMUM OF REAL ROOTS
CALL ZVALUE(X1,X2,X3,ZMAX,ZMIN)
C Z=ZMAX
WRITE(22,49),Z
GOTO 11
C FOR R=0
20 CONTINUE
C WRITE(22,21)
21 FORMAT(' THERE ARE THREE REAL ROOTS(2 EQUAL) OF THE CUBIC EQTN
1. IN Z')
IF (QQ.GT.0.0) GOTO 666
QQ=-QQ
AX=-((QQ/2.0)**(1.0/3.0))
GOTO 667
666 AX=((QQ/2.0)**(1.0/3.0))
667 X1=(AX+AX)-(B1/3.0)
X3=-AX-(B1/3.0)
X2=X3
C FINDING THE MAX. * MIN. OF DIFFERENT REAL ROOTS
CALL ZVALUE(X1,X2,X3,ZMAX,ZMIN)
C Z=ZMAX
WRITE(22,12),X1,X2,X3
C WRITE(22,49),Z
GOTO 11
C R>0
30 CONTINUE
C WRITE(22,31)
31 FORMAT(' THE CUBIC EQUATION IN Z HAS ONE REAL ROOT & TWO COMPLEX
1 ROOTS')
C CALCULATION OF THE ONE REAL ROOT
AX=(-(QQ/2.0)+SQRT(RR))
IF (AX.GT.0.0) GOTO 668
AX=-AX
AX=-AX**((1.0/3.0))
GOTO 669
668 AX=(AX**((1.0/3.0))
669 BX=(-(QQ/2.0)-SQRT(RR))
IF (BX.GT.0.0) GOTO 670
BX=-BX
BX=-BX**((1.0/3.0))
GOTO 671

```

```

670 BX=(BX**(.1,0/3,0))
671 Z=AX+BX-(B1/3,0)
IF(NO.NE.0) GO TO 100
ZMAX=Z
C THE REAL ROOT IS ASSIGNED TO ZMAX OR ZMIN DEPENDING ON WHETHER
C THE CALCULATION IS DONE FOR THE VAPOR OR THE LIQUID PHASE. THE
C OTHER ROOT IS ASSIGNED A HYPOTHETICAL VALUE AS THIS IS NOT
C USED SUBSEQUENTLY.
GIVE MINIMUM VALUE HYPOTHETICAL-10
ZMIN=-10.0
GO TO 200
100 CONTINUE
ZMIN=Z
C GIVE MAXIMUM VALUE HYPOTHETICAL 10
ZMAX=10.0
200 CONTINUE
C WRITE(22,49),Z
122 FORMAT(' ONE REAL ROOT=',F10.7)
11 CONTINUE
49 FORMAT(' THE VALUE OF Z TAKEN :',F15.6)
GO TO(3001,3001,3002)KSITU
3001 FIN=(Z**3.0)-Z*Z+Z*(ASTAR-BSTAR-BSTAR*BSTAR)-ASTAR*BSTAR
GO TO 3003
3002 FIN=(Z**3.0)+B1*Z*Z+C1*Z+D1
3003 CONTINUE
C WRITE(22,1621) FIN
1621 FORMAT(5X,'FUNCTION=',E16.9)
RETURN
END

```

```

C SUBROUTINE ZVALUE(X1,X2,X3,ZMAX,ZMIN)
C *****

```

```

C THIS SUBROUTINE CALCULATES THE MAXIMUM AND MINIMUM OF THREE
C REAL NO. X1,X2,X3

```

```

C SUBROUTINE ZVALUE(X1,X2,X3,ZMAX,ZMIN)

```

```

C ALL THREE POSITIVE
C IF ((X1.GT.0.0).AND.(X2.GT.0.0).AND.(X3.GT.0.0)) GO TO 10
C ALL THREE NEGATIVE
C IF ((X1.LT.0.0).AND.(X2.LT.0.0).AND.(X3.LT.0.0)) GO TO 20
C IF TWO ARE POSITIVE
C IF ((X1*X2*X3).LT.0.0) GO TO 30
C CASE4:1 POSITIVE, 2 NEGATIVE ROOTS
C ZMAX=AMAX1(X1,X2,X3)
C ZMIN=ZMAX
C GO TO 5
C CASE1:ALL POSITIVE

```

```

10 ZMAX=AMAX1(X1,X2,X3)
ZMIN=AMIN1(X1,X2,X3)
GO TO 5

```

```

C CASE2:ALL NEGATIVE
20 WRITE(22,7)
7 FORMAT(5X,'ERROR-ALL Z NEGATIVE')
GO TO 5

```

```

C CASE3:TWO POSITIVE & ONE NEGATIVE

```

```

30 IF ((X1*X2).LT.0.0) GO TO 40
A1=X1
A2=X2
GO TO 50
40 IF ((X1*X3).LT.0.0) GO TO 60
A1=X1
A2=X3
GO TO 50
60 A1=X2
A2=X3
50 ZMAX=AMAX1(A1,A2)
ZMIN=AMIN1(A1,A2)

```

```

5 CONTINUE
RETURN
END

```

```

C *****
C SUBROUTINE ZOOT(ZMAX,ZMIN,Y,F,B,FM,TC,PC,T,P,OMEGA,Z,N,KSITU)
C DIMENSION Y(25),TC(25),PC(25),OMEGA(25),TR(25),PR(25),B(25)
C DIMENSION TEMP(25),F(25)
C THIS SUBROUTINE GIVES Z GIVEN T,P,Y USING SRK OR PR FOR ONE
C PHASE, VAPOUR ALONE. Z IS THE CORRECT COPRESSIBILTY FACTOR
R=1.987

```

```

GA=0.4274803
BM=0.0
FM1=0.0
DEN=0.0
DO 1 I=1,N
TR(I)=T/TC(I)
PR(I)=P/PC(I)
B(I)=GB*R*TC(I)/(PC(I))
BM=BM+Y(I)*B(I)
TEMP(I)=(1.0+(0.480+1.574*OMEGA(I)-0.176*OMEGA(I)*OMEGA(I))*(1.0
1-((TR(I))*0.5)))
F(I)=(1.0/TR(I))*((TEMP(I))*2.0)
CONTINUE
DO 2 I=1,N
DO 3 J=1,N
FM1=Y(I)*Y(J)*(((TC(I)*F(I)*TC(J)*F(J))/(PC(I)*PC(J)))*0.5)+FM1
CONTINUE
DEN=DEN+((Y(I)*TC(I))/PC(I))
CONTINUE
FM=(FM1/DEN)
BSTAR=(BM*P)/(R*T)
ASTAR=((BSTAR*GA)/GB)*FM
B1=-1.0
C1=(ASTAR-BSTAR-(BSTAR*BSTAR))
D1=-(ASTAR*BSTAR)
GO TO 1004

```

```

1001 BM=0.
FM=0.
DO 1002 I=1,N
TR(I)=T/TC(I)
B(I)=0.07780*R*TC(I)/PC(I)
BM=BM+B(I)*Y(I)
ALPHA1=1.+(0.37464+1.54226*OMEGA(I)-0.26992*OMEGA(I)*OMEGA(I)
1)*((1-((TR(I))*0.5)))
ALPHA1=ALPHA1*ALPHA1
F(I)=(0.45724*((R*TC(I))*2)/PC(I))*ALPHA1
CONTINUE
DO 1003 I=1,N
DO 1003 J=1,N
FM=FM+Y(I)*Y(J)*((F(I)*F(J))*0.5)
ASTAR=FM*P/(R*T*R*T)
BSTAR=BM*P/(R*T)
B1=BSTAR-1.0
C1=ASTAR-3.0*(BSTAR*BSTAR)-2.0*BSTAR
D1=-BSTAR*(ASTAR-BSTAR-(BSTAR*BSTAR))
PP=(3.0*C1-(B1*B1))/3.0
QQ=(27.0*D1-9.0*B1*C1+2.0*B1*B1*B1)/27.0
RR=(PP*PP*PP)/27.0+QQ*QQ/4.0
IF ((ABS(RR)).GT.0.1E-10) GOTO 155
RR=0.0

```

```

1002 CONTINUE
DO 1003 I=1,N
DO 1003 J=1,N
1003 FM=FM+Y(I)*Y(J)*((F(I)*F(J))*0.5)
ASTAR=FM*P/(R*T*R*T)
BSTAR=BM*P/(R*T)
B1=BSTAR-1.0
C1=ASTAR-3.0*(BSTAR*BSTAR)-2.0*BSTAR
D1=-BSTAR*(ASTAR-BSTAR-(BSTAR*BSTAR))
PP=(3.0*C1-(B1*B1))/3.0
QQ=(27.0*D1-9.0*B1*C1+2.0*B1*B1*B1)/27.0
RR=(PP*PP*PP)/27.0+QQ*QQ/4.0
IF ((ABS(RR)).GT.0.1E-10) GOTO 155
RR=0.0

```

```

155 IF (RR) 10,20,30
CONTINUE
10 FORMAT(' THERE ARE THREE UNEQUAL ROOTS OF THE CUBIC EQTN. IN Z')
95 XS=((QQ*QQ/4.0)/((-PP*PP*PP)/27.0))
XS=SQRT(XS)
PHI=ACOS(XS)
AMP=2.0*((-PP/3.0))*0.5)
ANG=(PHI/3.0)
X1=AMP*COS(ANG)
X2=AMP*COS(ANG+2.094395)
X3=AMP*COS(ANG+4.18879)
IF (QQ.LT.0.0) GOTO 111
X1=-X1
X2=-X2
X3=-X3

```

```

111 X1=X1-(B1/3.0)
X2=X2-(B1/3.0)
X3=X3-(B1/3.0)
12 FORMAT(' ROOTS ARE:',10X,'X1=',F10.6,3X,'X2=',F10.6,3X,'X3=',F10
1.6)
CALL ZVALUE(X1,X2,X3,ZMAX,ZMIN)
Z=ZMAX
GOTO 11

```

```

20 CONTINUE
WRITE(50,21)
21 FORMAT(' THERE ARE THREE REAL ROOTS(2 EQUAL) OF THE CUBIC EQTN
1. IN Z')
IF (QQ.GT.0.0) GOTO 666
QQ=-QQ
AX=-((QQ/2.0))*((1.0/3.0))
GOTO 667

```

```

666 AX=((QQ/2.0))*((1.0/3.0))
667 X1=(AX+AX)-(B1/3.0)
X3=-AX-(B1/3.0)
X2=X3
CALL ZVALUE(X1,X2,X3,ZMAX,ZMIN)
Z=ZMAX

```

```

30 CONTINUE
31 FORMAT(' THE CUBIC EQUATION IN Z HAS ONE REAL ROOT & TWO COMPLEX
1 ROOTS.')
AX=(-(Q0/2.0)+SQRT(RR))
IF (AX.GT.0.0) GOTO 668
AX=-AX
AX=-(AX**(1.0/3.0))
GOTO 669
668 AX=(AX**(1.0/3.0))
669 BX=(-(Q0/2.0)-SQRT(RR))
IF (BX.GT.0.0) GOTO 670
BX=-BX
BX=-(BX**(1.0/3.0))
GOTO 671
670 BX=(BX**(1.0/3.0))
671 Z=AX+BX-(B1/3.0)
ZMAX=Z
ZMIN=-10.0
122 FORMAT(' ONE REAL ROOT=',F10.7)
11 CONTINUE
49 FORMAT(' THE VALUE OF Z TAKEN :',F15.6)
FIN=(Z**3.0)+(B1*Z*Z)+(C1*Z)+D1
RETURN
END
C*****
SUBROUTINE ENTHLP(P,T,Z,TC,PC,ZC,CPA,CPB,CPC,CPD,HACT,DELH,CPV)
PO=1.0
TO=298.0
R=1.987
VO=(R*T/PO)
V=(R*T*Z)/P
GA=0.4274802327
GB=0.086640350
A=GA*R*R*(TC**2.5)/PC
B=GB*R*TC/PC
TERM1=R*A*LOG((V-B)/V)
TERM2=(A/(B*SQRT(T)))*LOG((V+B)/V)
TERM3=R*A*LOG(V/VO)
DELA=-(TERM1*T)-TERM2-(T*TERM3)
DELS=TERM1-(TERM2/(2*T))+TERM3
DELH=DELA+(T*DELS)+(R*T*(Z-1.0))
HO=CPA*(T-TO)+((CPB/2.0)*(T*T-TO*TO))+((CPC/3.0)*(T*T*T-TO*TO*TO))
HACT=DELH+HO
CPV=CPA+CPB*T+CPC*T*T+CPD*T*T*T
RETURN
END
C *****
SUBROUTINE ENTL2(TC,PC,T,P,CPA,CPB,CPC,CPD,OMEGA,CPL,DELHVT,
1 DELHVT)
TR=T/TC
WRITE(22,100) T,TC,TR
100 FORMAT(5X,'T=',E15.8,3X,'TC=',E15.8,3X,'TR=',E15.8)
PR=P/PC
CPQ=CPA+CPB*T+CPC*T*T+CPD*T*T*T
R=1.987
TERM1=(0.5+2.2*OMEGA)
TERM2=(1.0-TR)**4
EXPR=TERM1*(3.67+(11.64*TERM2)+(0.634/(1.0-TR)))
CPL=CPQ+(EXPR*R)
TERM3=(1.0-TR)**0.354
TERM4=(1.0-TR)**0.456
DELHVT=((7.08*TERM3)+(10.95*OMEGA*TERM4))*R*TC
DELH=-DELHVT-(CPL*(298.0-T))
RETURN
END
C *****

```


 ** :::: APPENDIX - G :::: (Output listings) **

 OUTPUT LISTING FOR PROBLEM 4.1 IN CHAPTER 5

STAGE NO.	COMP. NO.	l(i,j) lb. moles/hr	v(i,j) lb. moles/hr	r(i) degree F
1	1	.107667E+03	.215333E+02	120.0000
	2	.532210E+01	.106442E+01	
	3	.111667E-01	.223335E-02	
2	1	.108856E+03	.129200E+03	137.7383
	2	.117392E+02	.638652E+01	
	3	.614667E-01	.134001E-01	
3	1	.963020E+02	.130390E+03	144.0902
	2	.202026E+02	.128036E+02	
	3	.246568E+00	.637001E-01	
4	1	.806773E+02	.117835E+03	152.8198
	2	.304112E+02	.212671E+02	
	3	.853805E+00	.248801E+00	
5	1	.636820E+02	.102211E+03	163.8070
	2	.401304E+02	.314756E+02	
	3	.255316E+01	.856038E+00	
6	1	.473355E+02	.852153E+02	177.0408
	2	.461006E+02	.411948E+02	
	3	.650282E+01	.255540E+01	
7	1	.333305E+02	.688689E+02	193.0444
	2	.455351E+02	.471650E+02	
	3	.138394E+02	.650505E+01	
8	1	.509205E+02	.548639E+02	212.2850
	2	.859456E+02	.465995E+02	
	3	.543537E+02	.138416E+02	
9	1	.453253E+02	.494538E+02	216.3888
	2	.910728E+02	.500100E+02	
	3	.552782E+02	.143560E+02	
10	1	.388896E+02	.438586E+02	221.3394
	2	.971532E+02	.551373E+02	
	3	.564654E+02	.152804E+02	
11	1	.319247E+02	.374229E+02	227.0275
	2	.103814E+03	.612176E+02	
	3	.580314E+02	.164677E+02	
12	1	.248646E+02	.304580E+02	233.2853
	2	.110223E+03	.678788E+02	
	3	.602568E+02	.180336E+02	
13	1	.181735E+02	.233979E+02	240.0673
	2	.114870E+03	.742878E+02	
	3	.637971E+02	.202591E+02	
14	1	.122480E+02	.167069E+02	247.7561
	2	.115326E+03	.789342E+02	
	3	.700152E+02	.237993E+02	
15	1	.736771E+01	.107814E+02	257.4983
	2	.108269E+03	.793901E+02	
	3	.812966E+02	.300174E+02	
16	1	.146665E+01	.590106E+01	271.2453
	2	.359356E+02	.723338E+02	
	3	.399978E+02	.412988E+02	

STAGE NO.	COMP. NO.	l(i,1) lb.moles/hr	v(i,1) lb.moles/hr	T(i) degrees F
1	1	.107366E+03	.214732E+02	120.0000
	2	.559345E+01	.111869E+01	
	3	.403065E-01	.806130E-02	
2	1	.100266E+03	.128839E+03	131.3184
	2	.130268E+02	.671213E+01	
	3	.194119E+00	.483678E-01	
3	1	.836043E+02	.121739E+03	140.6481
	2	.237202E+02	.141454E+02	
	3	.693419E+00	.202180E+00	
4	1	.643340E+02	.105078E+03	154.3571
	2	.359019E+02	.248389E+02	
	3	.203641E+01	.701480E+00	
5	1	.467383E+02	.858072E+02	170.6858
	2	.456364E+02	.370206E+02	
	3	.494481E+01	.204447E+01	
6	1	.333734E+02	.682115E+02	187.0100
	2	.496256E+02	.467551E+02	
	3	.100675E+02	.495287E+01	
7	1	.243485E+02	.548467E+02	202.0558
	2	.470584E+02	.507443E+02	
	3	.175051E+02	.100756E+02	
8	1	.409616E+02	.458218E+02	215.9916
	2	.870244E+02	.481771E+02	
	3	.581255E+02	.175142E+02	
9	1	.355622E+02	.394348E+02	220.5677
	2	.920391E+02	.511431E+02	
	3	.594876E+02	.181335E+02	
10	1	.299416E+02	.340354E+02	225.4205
	2	.972974E+02	.561578E+02	
	3	.610574E+02	.194957E+02	
11	1	.243634E+02	.284149E+02	230.3501
	2	.102339E+03	.614161E+02	
	3	.629097E+02	.210654E+02	
12	1	.190779E+02	.228367E+02	235.2433
	2	.105499E+03	.664582E+02	
	3	.652636E+02	.229177E+02	
13	1	.142774E+02	.175511E+02	240.4625
	2	.103791E+03	.706172E+02	
	3	.695111E+02	.252717E+02	
14	1	.100829E+02	.127506E+02	245.4670
	2	.107755E+03	.729094E+02	
	3	.739010E+02	.286191E+02	
15	1	.855301E+01	.855610E+01	251.9287
	2	.101333E+03	.718738E+02	
	3	.827046E+02	.339091E+02	
16	1	.152675E+01	.502626E+01	260.7984
	2	.358813E+02	.654521E+02	
	3	.399919E+02	.427127E+02	

OUTPUT LISTING FOR PROBLEM 4.3 IN CHAPTER : 5

STAGE NO.	COMP. NO.	l(i,1) lb.moles/hr	v(i,1) lb.moles/hr	T(i) degrees F
1	1	.104041E+03	.208083E+02	120.0000
	2	.891244E+01	.178249E+01	
	3	.460705E-01	.921411E-02	

3	1	.102195E+03	.124850E+03	151.6332
	2	.187873E+02	.106949E+02	
	3	.239171E+00	.552846E-01	
4	1	.852580E+02	.123003E+03	163.4995
	2	.298592E+02	.205698E+02	
	3	.867791E+00	.248385E+00	
5	1	.665745E+02	.106066E+03	177.2026
	2	.405800E+02	.316416E+02	
	3	.264118E+01	.877005E+00	
6	1	.486491E+02	.873827E+02	193.9641
	2	.475190E+02	.424625E+02	
	3	.675657E+01	.265040E+01	
7	1	.335807E+02	.694574E+02	213.6982
	2	.475276E+02	.494015E+02	
	3	.143495E+02	.676578E+01	
8	1	.495558E+02	.543890E+02	219.2165
	2	.894564E+02	.494101E+02	
	3	.552164E+02	.143587E+02	
9	1	.421788E+02	.473640E+02	225.8087
	2	.959759E+02	.542389E+02	
	3	.566989E+02	.152256E+02	
10	1	.340672E+02	.399871E+02	233.4754
	2	.102890E+03	.607583E+02	
	3	.589102E+02	.167081E+02	
11	1	.257525E+02	.318755E+02	242.5076
	2	.108755E+03	.676728E+02	
	3	.625457E+02	.189194E+02	
12	1	.178480E+02	.235608E+02	253.8756
	2	.110937E+03	.735371E+02	
	3	.690441E+02	.225549E+02	
13	1	.109513E+02	.156563E+02	269.3115
	2	.105645E+03	.757195E+02	
	3	.809097E+02	.290534E+02	
	1	.219171E+01	.875957E+01	
	2	.352175E+02	.704270E+02	
	3	.399908E+02	.409189E+02	

OUTPUT LISTING FOR PROBLEM 5.1 IN CHAPTER : 5

STAGE	COMP.	l(i,j)	v(i,j)	T(i)
NO.	NO.	lb. moles/hr	lb. moles/hr	degree F
1	1	.107691E+03	.215383E+02	120.0000
2	2	.529540E+01	.105908E+01	137.7123
	3	.132378E-01	.264756E-02	
	1	.108883E+03	.129230E+03	
3	2	.115816E+02	.635448E+01	3.2438(=M(i))
	3	.723798E-01	.158853E-01	
	1	.621503E+02	.130421E+03	
4	2	.411949E+02	.127406E+02	178.0969
	3	.262032E+01	.755273E-01	
	1	.461548E+02	.836885E+02	
5	2	.463044E+02	.422540E+02	193.9409
	3	.660665E+01	.262297E+01	
	1	.325838E+02	.676931E+02	
6	2	.460384E+02	.479634E+02	212.9258
	3	.139567E+02	.660930E+01	
	1	.500626E+02	.541221E+02	
7	2	.865683E+02	.470974E+02	217.2546
	3	.545699E+02	.139593E+02	
	1	.442759E+02	.486009E+02	
	2	.915433E+02	.506274E+02	
	3	.555945E+02	.145725E+02	
	1			

	1	.375243E+02	.428142E+02	
	2	.972784E+02	.557023E+02	
9	3	.573746E+02	.156971E+02	3.8132(=M(i))
	1	.122085E+02	.361626E+02	
	2	.115354E+03	.613375E+02	
10	3	.700195E+02	.173772E+02	257.5227
	1	.734338E+01	.107468E+02	
	2	.108290E+03	.794130E+02	
11	3	.812992E+02	.300222E+02	271.2580
	1	.146173E+01	.588165E+01	
	2	.359409E+02	.723495E+02	
	3	.399974E+02	.413019E+02	

OUTPUT LISTING FOR PROBLEM 5.2 IN CHAPTER : 5

STAGE	COMP.	l(i,i) lb.moles/hr	v(i,i) lb.moles/hr	T(i) degrees F
NO.	NO.			
1	1	0.104057E+03	0.208113E+02	120.0
	2	0.889685E+01	0.177937E+01	
	3	0.462505E-01	0.925009E-02	
2	1	0.102218E+03	0.124868E+03	142.5902
	2	0.187571E+02	0.106752E+02	
	3	0.240149E+00	0.555006E-01	
3	1	0.667419E+02	0.123030E+03	2.0(=M(i))
	2	0.405585E+02	0.205365E+02	
	3	0.262996E+01	0.249399E+00	
4	1	0.487864E+02	0.875533E+02	177.0767
	2	0.475263E+02	0.423378E+02	
	3	0.673640E+01	0.263921E+01	
5	1	0.336739E+02	0.695977E+02	193.8358
	2	0.475731E+02	0.493056E+02	
	3	0.143213E+02	0.674555E+01	
6	1	0.497244E+02	0.544853E+02	213.6982
	2	0.894976E+02	0.493524E+02	
	3	0.552074E+02	0.143306E+02	
7	1	0.423330E+02	0.475358E+02	219.4111
	2	0.960456E+02	0.542769E+02	
	3	0.566768E+02	0.152166E+02	
8	1	0.342073E+02	0.401444E+02	225.6880
	2	0.103021E+03	0.608250E+02	
	3	0.598540E+02	0.166851E+02	
9	1	0.178540E+02	0.320187E+02	2.022(=M(i))
	2	0.111083E+03	0.678000E+02	
	3	0.690891E+02	0.188633E+02	
10	1	0.109502E+02	0.156654E+02	253.8810
	2	0.105773E+03	0.758630E+02	
	3	0.809789E+02	0.290984E+02	
11	1	0.218862E+01	0.876159E+01	269.3195
	2	0.352206E+02	0.705519E+02	
	3	0.399991E+02	0.409882E+02	
